UIUCDCS-R-76-821

INTERACTIVE TEST CONSTRUCTION AND ADMINISTRATION IN THE GENERATIVE EXAM SYSTEM

by

Lawrence Robert Whitlock

August 1976

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-76-821

INTERACTIVE TEST CONSTRUCTION AND ADMINISTRATION IN THE GENERATIVE EXAM SYSTEM

by

Lawrence Robert Whitlock

August 1976

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

Supported in part by the National Science Foundation under grant number NSF EPP 7421590.

ACKNOWLEDGEMENT

The author wishes to express his gratitude to Professor Wilfred J. Hansen for his many suggestions and guidance throughout the work on this thesis. Appreciation is also in order to Professor R. Montanelli and Professor J. Nievergelt for their helpful suggestions on this project.

TABLE OF CONTENTS

1. INTRODUCTION

The Generative Exam System is a completely interactive system for construction and administration of examinations. During a single terminal session, the system can administer an examination, grade it, and allow the student to compare his answers with the correct ones. An exam consists of several "problems" each administered by an independent problem generator/grader (pg/g) module according to specifications written by the instructor. Analyses of student performance, class performance, and examinations are also provided by the system.

This paper describes the implementation problems and solutions for the Generative Exam System and compares testing via the system with the traditional form of testing--written exams. The Generative Exam System provides advantages over written exams such as ease in test construction, interactive test administration, objective grading, immediate feedback of exam results for the student, automatic record keeping, fast analyses of exam results, and a variety of displays of exam results and analyses. Studies comparing exams administered by the Generative Exam System with written exams indicate that the computer exams are as effective at evaluating students as written exams. Chapter 2 of this paper describes the logical structure of the exam system. Problem generation and grading schemes are discussed in Chapter 3. Chapter 4 outlines the experiments conducted to aid in the development of the system. The studies of the effectiveness of testing with the Generative Exam System are described in Chapter 5.

With the capabilites of the Generative Exam System, it became plausible to study the idea of a "tailored" exam. A tailored exam attempts to administer to each student questions which are of a difficulty suited to his level of knowledge. Studies of tailored exams indicate that the tailoring idea is effective, but the approach used in the Generative Exam System to tailor an exam is inefficient and unpopular. An alternate approach to tailoring is proposed which would be more efficient and might eliminate some of the unpopularity of the tailored exam. These ideas and studies are discussed in Chapter 6.

1.1 BACKGROUND

Several factors motivated the construction of the Generative Exam System. The Department of Computer Science has been working on a project to partially automate the introductory computer science courses (20) by developing a subsystem for computer science instruction on the PLATO IV Computer-based Education system (19, 29) at the University of Illinois. An exam system was needed to round out the usefulness of this automated instruction system.

An exam system could also be useful independently since it would save considerable time and expense in writing, duplicating, and grading of exams. Further, better exams could be prepared through the exam system since a large library of tested exam problems would be available. Since exams are easily written in the exam system, more exams could be given which could lead to better evaluations of students.

Better evaluations could also be achieved through improved problem generator/graders. As they became more sophisticated they could assign grades on more information than just answer correctness. Other factors that could also be used include the length of time the student spent on the problem, the number of times he changed his answers, the amount of use he made of any on-line references (eg. a dictionary of terms), and the algorithm used. This score might be more indicative of a student's knowledge of the material than the number of correct responses.

The computerized exam system could also provide a convenient environment for experimentation with other styles of exams and other means of evaluating students.

Lippey (17) has described many areas in the expanding field of Computer Assisted Test Construction. Many currently used test construction systems produce printed tests from large item pools (2, 3, 5, 6, 11, 13, 15, 24, 28). Other systems produce printed tests from item generators (14, 21, 23, 30). Computer constructed tests are used in many Computer Managed Instruction systems (10, 26, 27).

McClain (18) describes a system which constructs exams from item pools, grades answer sheets, and analyses exam results. An item pool is maintained for each subject (eg. chemistry). The system can produce Coursewriter III code for administering the exam interactively from a terminal. The system also has the capability of generating multiple choice questions.

The Generative Exam System goes beyond these systems in several ways. Convenience is provided by the fact that all activities on the exam system are interactively conducted from a terminal. The problem generator/graders are independent which permits the use of question styles other than the usual multiple choice, true-false, or matching style questions. More sophisticated generation schemes are used to produce a great variety of questions from each pg/g. Grading schemes are employed which award partial credit for answers that are partially correct. The Generative Exam System has also provided an environment for experimenting with non-traditional styles of testing (eg. the tailored exam).

1.2 ENVIRONMENT

The Generative Exam System is implemented on the PLATO IV Computer-based Education system (19, 29). PLATO is a large system capable of servicing up to 1000 terminals. The PLATO terminal uses a plasma panel display on which can be displayed 32 lines of test 64 characters wide at a rate of 180 characters per second. It also has graphic capabilities and can draw 60 lines per second. Input is usually through a keyboard which consists of a standard typewrite set of keys plus several special function keys (eg. NEXT, BACK, HELP, DATA, STOP).

Programs in PLATO are referred to as "lessons".

Three levels of physical memory are used in the PLATO system: all lessons and data are permanently stored on disc; active lessons and data are held in a large auxilliary core memory; and the lesson and data being used by the student at the currently active terminal are stored in the computer's central memory. When a student begins a session at a terminal, his data and the lesson he selects are transferred from disc to the auxilliary memory. For each of his timeslices, his lesson and data are transferred into central memory at the beginning, and back to the auxilliary memory at the end. When a student finishes his session at a terminal, his data is transferred from the auxilliary memory to disc.

Work on the Generative Exam System began in early 1975, and the first exam using the system was administered in the summer of that year. Several exams have been administered by the system in the year since that first exam.

2. SYSTEM DESCRIPTION

The Generative Exam System provides a user with facilites for taking an exam, reviewing his last exam, and resuming work in his last exam. The system also provides an instructor with facilites for writing exams, seeing displays and analyses of data collected from exams, and other system maintenance tasks.

A detailed description of the Generative Exam System is given in another document (34), but it is briefly outlined below. Figure 2.1 shows a block diagram of the system. The heart of the system is the set of problem generator/grader (pg/g) modules. Each pg/g carries out all facets of administering problems over a small set of concepts except for data storage. The remainder of the exam system handles the data storage and analysis and the routing of the user to the appropriate sections or pg/g's in the system. The exam system is designed to handle up to 1000 students.

2.1 DATA

Three data bases are maintained by the exam system. The data contained in each is briefly described below.

2.1.1 EXAM SPECIFICATIONS DATA BASE

An exam specification is a set of problem specifications plus exam identification information. Problem specifications are written by the instructor in each pg/g used in his exam, and these specifi-

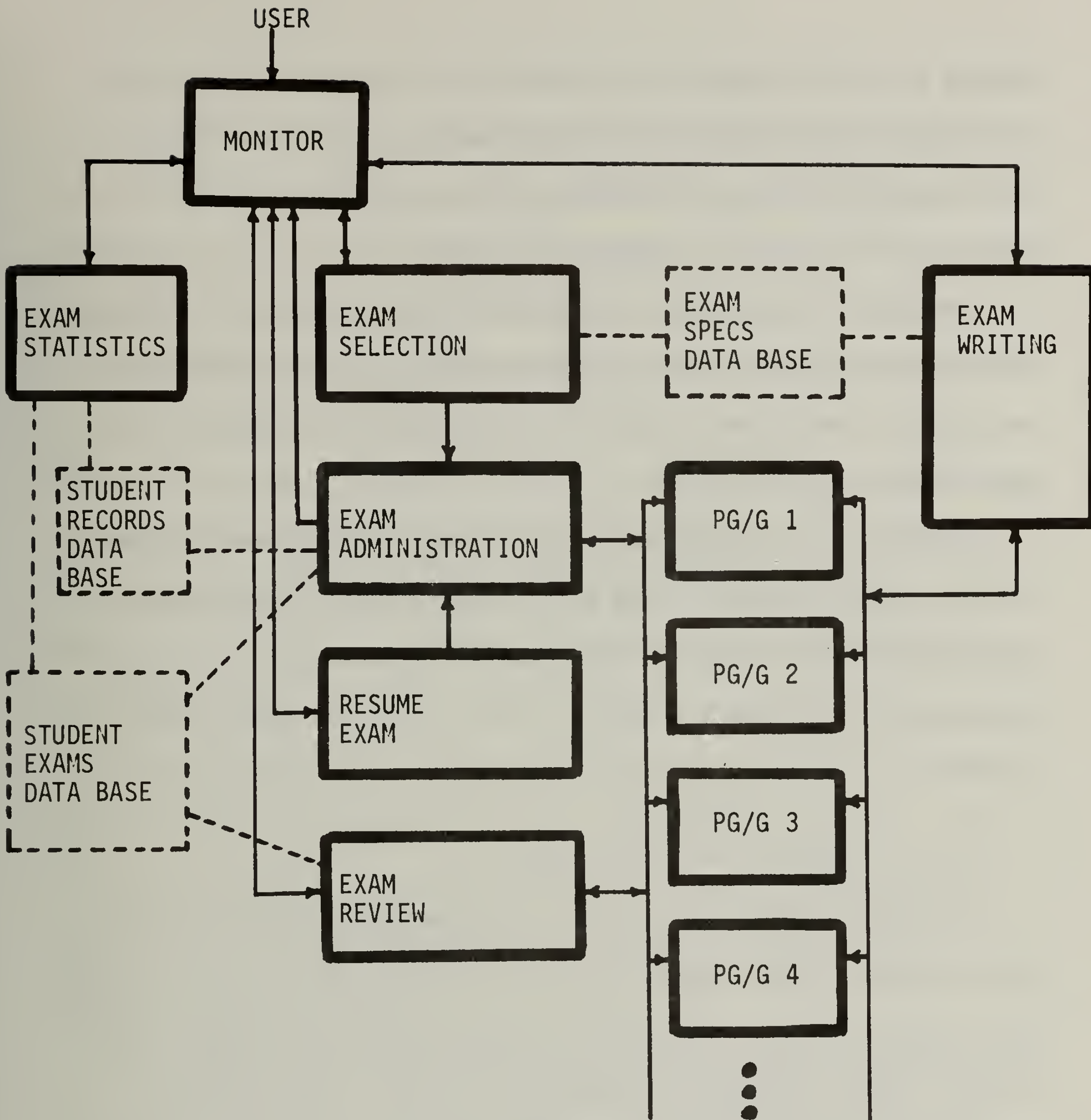
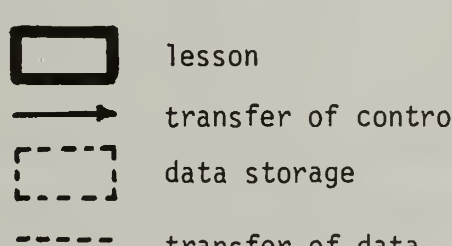


FIGURE 2.1: BLOCK DIAGRAM OF THE GENERATIVE EXAM SYSTEM



cations guide the generation of the questions in the problem. The exam identification information specifies, among other things, the course to which the exam is available and whether the exam is a practice exam or is to be taken for a grade.

The exam specifications are stored in the Exam Specs Data Base. When a student takes an exam, this data base is accessed for exams available to the student's course. If an exam is available, it is administered to the student.

When an exam is selected for the student, a copy of the exam specification is stored in the user's Student Exam record where it guides the administration of the exam. This structure of the exam system permits different students to take different exams concurrently.

2.1.2 STUDENT EXAMS DATA BASE

A Student Exam record is an area on permanent storage (disc) where the user's exam specification and work on that exam are stored. The record is large enough to hold only one exam at a time, so only the last exam a user took is kept by the system. When taking an exam, each time the student finishes working on a problem, his work for that problem is transferred to his Student Exam record on disc. This is done to insure that his work is not lost in the event of a PLATO system failure or an accidental press of the keys SHIFT-STOP. (SHIFT-STOP is the signal to the PLATO system that the student wants to immediately sign off from his terminal.)

Frequent disc accesses are discouraged by the PLATO staff since a high demand on the disc controllers by one PLATO user might cause annoying delays in service to other PLATO users. For this reason the Generative Exam System originally stored each student's exam specification and work (i.e. his Student Exam) in the auxilliary memory. However, the auxilliary memory is only a termporary storage area and difficulties were encountered in recovering Student Exams after a PLATO system failure. Further, since the amount of space in the auxilliary memory was limited for each room of PLATO terminals, storing Student Exams in the auxilliary memory created a greater demand for space in the auxilliary memory than was allocated to the room of terminals.

The best solution to these problems was to store the Student Exams on disc. The only time a Student Exam occupies space in the auxilliary memory is when a student's latest work on a problem is copied into his Student Exam (i.e. each time the student leaves a problem to work on another).

The PLATO staff estimated that an average of one disc access per minute with a burst rate of less than five per minute would probably be acceptable. The Generative Exam System requires about 15 to 30 disc accesses per student for a five-problem exam lasting one hour. This is well within the estimated limits.

2.1.3 STUDENT RECORDS DATA BASE

Each user is assigned a Student Record in which is recorded user identification information and summary information for the last exam he has taken (scores, times, etc.). When a student finishes his exam, the necessary information is copied from his Student Exam into his Student Record.

When an instructor chooses to see information about the performance of a class on an exam, data is collected from the appropriate Student Records, analysed, and displayed. Student Records are maintained so that no disc accesses are required for analyses of exam results. This makes rapid data analysis and presentation possible.

2.1.4 DATA SECURITY

All lesson source code and data storage areas in the Generative Exam System are protected by the PLATO password system. Only users who can correctly enter the assigned passwords are permitted to access the source code and data storage areas.

2.2 USER INTERACTION

The Generative Exam System differentiates between two types of users--student and instructor. The features available to each user type are outlined below.

2.2.1 STUDENT OPTIONS

A student has four options in the Generative Exam System: take an examination for a grade; take a practice exam; resume working in the last exam he was taking; or look at the scores and answers on his last exam.

The only difference between taking an exam for a grade and taking a practice exam is that after an exam for a grade, the student is not permitted to take another exam or resume working in his last exam until the instructor resets a permission flag in the student's Student Record. Since only the last exam the student took is stored in the system, this restriction is put on students after taking an exam for a grade so that the instructor can collect data on one exam before the student takes another.

2.2.2 INSTRUCTOR OPTIONS

An instructor has access to all of the student options plus six other options: write or modify an examination; see a graph of student data; see a list or make a print of student data; see a student's record or his exam; change students' permission for exam access; and delete students from the exam system.

To write an exam, an instructor selects problems from a list of available problem generator/graders and writes problem specifications in each pg/g. The sets of problem specifications are assembled together along with exam identification information specified by the instructor into an exam specification and stored in the Exam Specs Data Base for student use.

The instructor may see graphs of the distributions of the data collected from a group of students' exams. He may also have the data listed on the PLATO screen or printed out on paper.

Data in any student's exam or Student Record may be viewed and modified by the instructor. This facilitates hand grading and adjustment of scores in the event of an error in the system.

The instructor may alter any student's permission flag which changes the options available to that student. For example, through this facility, an instructor can permit a student to resume working on an exam which that student had taken for a grade.

Instructors may delete students from the exam system to make room for other students in the exam system's records. (A student is automaticaaly allocated a Student Record and a Student Exam record the first time he enters the exam system.)

2.3 PROBLEM GENERATOR/GRADERS

Each problem generator/grader is an independent module which handles all aspects of one problem except data storage. All data is handled by the exam system in such a fashion that each pg/g has free use of all storage areas available to a PLATO program. The modularity of the pg/g's permits great flexibility in the style of questions produced by the different pg/g's. Since each pg/g is not restricted to producing a particular style of question (eg. multiple choice questions) it can use the approach most appropriate to the concepts it tests. The simplicity of interfacing pg/g's to the exam system facilitates expansion of the problem repertoire.

2.3.1 GENERAL STRUCTURE

Each pg/g has five major sections: problem specifications writing section; administration section; review section; generation section; and evaluation section (see Figure 2.2).

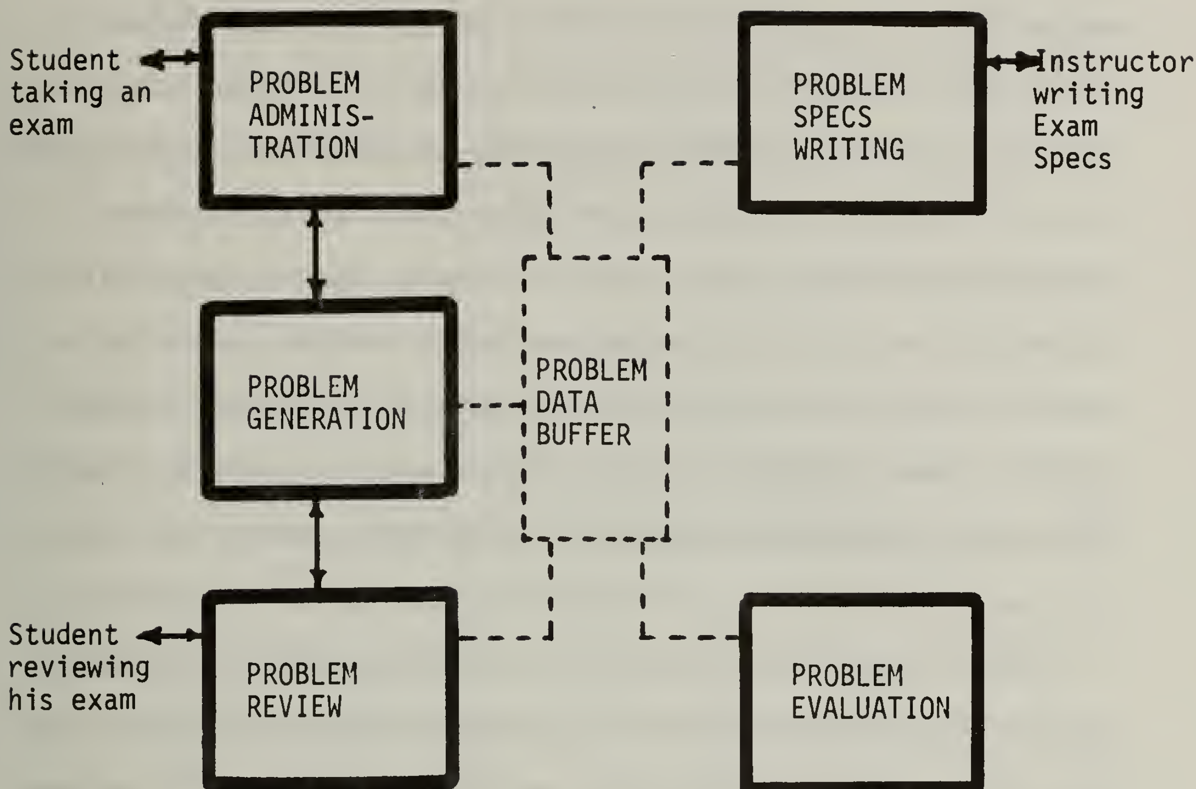
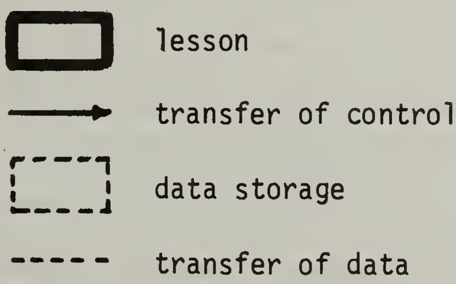


FIGURE 2.2: BLOCK DIAGRAM OF A PROBLEM GENERATOR/GRADER



The problem specifications writing section is accessed during exam writing. In this section, the instructor indicates what parts of the pg/g's capabilities should be used for his problem. For example, in a Fortran Expressions problem the instructor might choose to have precedence, parentheses, mixed-mode arithmetic, built-in functions, and integer division tested but not double exponentiation and unary minus. Facilities are provided so that the instructor may try sample problems generated according to his problem specifications. When the instructor is satisfied with the problem produced by the pg/g, the problem specifications for this problem are stored with the other problem specifications in the exam he is writing.

When a student, taking an exam, enters the pg/g administration section for the first time, the problem data buffer will contain the problem specifications which guide the generation of the problem for this student. After the problem has been generated and on subsequent entries into the pg/g administration section, the data buffer will contain problem parameters and the student's work in addition to the problem specifications. The pg/g administration section then displays the problem and any previous work the student did on this problem. New answers may then be received, stored, and graded. When the student chooses to leave the problem, the exam system stores the contents of the data buffer in the Student's Exam on disc.

The pg/g review section is accessed when the student reviews his exam. It receives the same problem specifications, problem parameters, and student's work in the data buffer as did the administration section. If the student did not work on this problem during his exam, a typical problem is generated at this point. The student's problem is displayed along with his responses, the correct answers, the scores earned, and any explanations that may help in understanding the display or his errors.

The generation section is accessed by the administration section and the review section. The generation section produces problem parameters from which a unique problem is presented to the student. These problem parameters are kept with the student's work on this problem so that he will receive the same set of questions each time he reenters this pg/g during an exam.

The evaluation section of a pg/g keeps statistics on problem use. These statistics are used by the pg/g author to improve the quality of the problems produced. The statistics would also be used for student comparisons when the pg/g is used by the Quiz System (1).

2.3.2 EXAMPLES OF PROBLEM GENERATOR/GRADERS

Fifteen pg/g's are currently available in the Generative Exam System. (See Appendix A for a complete listing of the pg/g's and their authors.) Some of the pg/g's are "tailoring" pg/g's. These generate a problem to a given level of difficulty in addition to the constraints specified by the instructor. When a "tailoring" pg/g is used in a tailored style exam (see Chapter 6), the system determines a difficulty level which is passed with the problem specifications to the pg/g. When a "tailoring" pg/g is used in a regular style exam, the difficulty level specified by the instructor when writing problem specifications is used.

The pg/g's on Fortran Expressions present expressions which the student evaluates (see Figure 2.3). The concepts that may be covered include precedence, double exponentiation, unary minus, built-in functions, partheneses, mixed-mode arithmetic, and integer division.

In a problem produced by the pg/g on Fortran PRINT with FORMAT, the student is shown a program segment consisting of some assignment statements, a PRINT statement, and a FORMAT statement (see Figure 2.4). He is required to show the output on a grid as it would appear on a printout. The problem covers I, F, and E format codes, slash, Hollerith strings, field counts, and group counts.

The pg/g on DO-loops Over an Expression shows the student a program segment consisting of a DO-loop which contains some calculations and a PRINT statement (see Figure 2.5). He is required to show what is printed by the program segment. The instructor may select either Fortran or PL/1 for the problem.

In the READ with FORMAT pg/g (see Figure 2.6), the student is required to show the exact values stored when executing a formatted READ statement. The problem displays an input data card from which the values are read.

The One-Dimension Fortran Array problem (see Figure 2.7) requires that the student work through a program which manipulates data in one-dimensional arrays. The student must show the initial contents of the arrays and the contents of the arrays at the end of

```
Problem 8                                                  38 points
--------------------------------------------------------------------
   Type in the value for each expression.  Assume default
        declarations for the variables.  Include a decimal
        point if and only if the value is real.

  For  FOW=-2.  PY=10.  VY=2.  calculate:
  3.5+FOW-40./PY/VY                                 >

  For  PY=30.  ZO=6.  VAS=9.5  calculate:
  (PY/(6.+ZO))/2.+VAS

  For  VAS=-7.  VY=1.  calculate:
  -VAS**2.-VY

  For  VY=-4.  PY=5.  VAS=4.  calculate:
  ABS(VY*0.**PY*VAS)

  For  FED=1.  PY=2.  VAS=-1.  calculate:
  10.**FED**PY-VAS

  For  MAF=-4  IIT=48  NE=6  calculate:
  MAF/((IIT/NE)-10)/2

  For  FED=2.  VY=1.  FOW=5.  calculate:
  FED**VY**0.-FOW

  For  ZO=-4.  VAS=0.  PY=-3.  calculate:
  -ZO**VAS-PY
--------------------------------------------------------------------
SHIFT-NEXT to next problem;  SHIFT-BACK to previous problem
SHIFT-DATA to return to the cover page;
```

FIGURE 2.3: TYPICAL FORTRAN EXPRESSIONS PROBLEM

Problem 0 25 points

Show exactly what is printed by this program segment.

```
      I=65
      J=4
      K=24
      L=33
      M=2
      PRINT20,I,J,K,L,M
   20 FORMAT('1',2I3/' GFBL',2(I3,2X))
```

```
    1 column 10        20        30        40        50        60
line↓        ↓         ↓         ↓         ↓         ↓         ↓
 1   ............................................................
 2   ............................................................
 3   ............................................................
 :
```

Type in the grid line number you want to write on: ≫

SHIFT-NEXT to next problem; SHIFT-BACK to previous problem
SHIFT-DATA to return to the cover page:

FIGURE 2.4: TYPICAL FORTRAN PRINT WITH FORMAT PROBLEM

```
Problem 0                                                    25 points
------------------------------------------------------------------------
Type in what this Fortran segment prints.


Enter "end" when there is no more output to be printed.
Enter "del" to delete an answer.


      INTEGER W4,Z6,A2,E7,Y3          | OUTPUT:   W4              Z6
      Y3 = 5                          |
      E7 = 30                         | >
      Z6 = 0                          |
      A2 = 3                          |
      DO  20  W4 = A2, E7, Y3         |
      IF (W4 .EQ. 13)  GOTO 20        |
      Z6 = Z6 + 2 * W4 + 4            |
      PRINT, W4, Z6                   |
20    CONTINUE                        |
30    CONTINUE                        |
------------------------------------------------------------------------
SHIFT-NEXT to next problem;   SHIFT-BACK to previous problem
SHIFT-DATA to return to the cover page;
```

FIGURE 2.5: TYPICAL DO-LOOPS OVER AN EXPRESSION PROBLEM

```
.. Problem 3          Read with format                    3Ø points

  Type in the value stored in each variable by the following
     program segment.  Include a decimal point if and only
     if the value is real.


      READ1Ø,MI,SEB,TOR,WI,FE
   1Ø FORMAT(I3,F6.2,2E8.1,1X,F6.3,1X,3X,2X)


          Input Card
   1 Column 1Ø          2Ø          3Ø          4Ø          5Ø          6Ø
   ↓        ↓           ↓           ↓           ↓           ↓           ↓
  /1475478205957319.Ø7655.49Ø238ØØ388334305

          MI=

          SEB=

          TOR=

          WI=

          FE=

SHIFT-NEXT to next problem;  SHIFT-BACK to previous problem
SHIFT DATA to return to the cover page;                             8
```

FIGURE 2.6:   TYPICAL READ WITH FORMAT PROBLEM

Problem 0   40 points

Show the values contained in array QM and array DH after executing statement 10 and after executing statement 30.

```
      INTEGER I,
     X  QM(5)/4*1,0/,
     X  DH(5)/3*3,2*0/
 10   CONTINUE
      I=1
 20   CONTINUE
      QM(I)=DH(I+1)+QM(6-I)
      I=I+1
      IF (I.LE.4) GO TO 20
 30   CONTINUE
```

Values in arrays at statement 10:

QM(1)	QM(2)	QM(3)	QM(4)	QM(5)

DH(1)	DH(2)	DH(3)	DH(4)	DH(5)

Values in arrays at statement 30:

QM(1)	QM(2)	QM(3)	QM(4)	QM(5)

DH(1)	DH(2)	DH(3)	DH(4)	DH(5)

SHIFT-NEXT to next problem; SHIFT-BACK to previous problem
SHIFT-DATA to return to the cover page;   10

FIGURE 2.7: TYPICAL ONE-DIMENSIONAL FORTRAN ARRAY PROBLEM

execution of the program segment.

In the pg/g for Short Answer Questions, the student is presented true/false, multiple choice, or fill-in-the-blanks questions. The questions available from this pg/g are written by the instructor and entered into the pg/g while he is writing problem specifications. In each question, he specifies items that can be generated by the pg/g. For each item that can be generated, he specifies the type (variable name, value, etc.) and the constraints for generation (maximum value, minimum value, mode, etc.). This pg/g permits expansion of the test item pool by instructors who do not want to write a problem generator/grader.

3. GENERATION AND GRADING SCHEMES

3.1 GENERATION

To insure exam security a large item pool is required. Prosser (24) suggests that ten times as many items are required in the item pool as will appear on any one test. Even when it is made available to students, a large item pool makes it impractical for them to attempt to just memorize the answers to the items in the pool.

The Generative Exam System does not have an explicit item pool but rather has a pool of problem generator/graders, each of which can produce a very large number of different problems. The item pool for the Generative Exam System is effectively unlimited. Not only does this eliminate the problem of security for test questions, it also encourages honesty during the administration of an exam because no two students receive identical sets of questions from any given pg/g.

3.1.1 GENERAL APPROACHES TO GENERATION

Three general approaches to generation are used in Computer Assisted Test Construction and Computer Assisted Instruction. One approach, common in Computer Assisted Test Construction systems, is the use of random numbers or randomly generated character strings (14, 18, 25, 30, 31, 32). Often the range of a randomly generated number is restricted so the problem makes sense or to coordinate it with previously generated numbers in the problem.

A second approach is the assembly of problem pieces into a complete structure (8, 16, 21, 25). The assembly process is controlled by a grammar or by selection from pools of problem pieces. The more complex schemes in this approach are found in Computer Assisted Instruction applications rather than test construction systems.

A third approach accesses an information network to flesh out question forms (7, 33). This approach is being researched in some Computer Assisted Instruction applications.

Problem generator/graders currently available in the Generative Exam System generally use a combination of the first and second approaches (see Section 3.1.3).

3.1.2 CONSTRAINTS ON PROBLEM GENERATOR/GRADERS

The generation schemes used in problem generator/graders are constrained by several design factors:

> The content of the questions produced by a pg/g should be specifiable by the instructor. For example, in a problem on expressions, the instructor may want to test the peculiarities of double exponentiation but not unary minus.
>
> Each question in a problem should test something significant and unique from the other questions. This is in contrast to drill exercises where repetition is desirable.
>
> The generation process should not take a long time. But, the amount of permanent storage space required by the pg/g should also be minimized.

Problems should not be so complex that entering answers is difficult and grading answers takes a long time.

The magnitude of the numbers used in the problem should be small enough to avoid long or complicated calculations.

Problems need to be designed so that they fit neatly on the screen. That is, strings of numbers or characters (eg. numbers on a data card) may need to be constrained so that they always fit into the problem display.

Finally, it is desirable that the pg/g be capable of generating questions to different levels of difficulty for use with non-traditional styles of exams (see Chapter 6).

3.1.3 GENERATION SCHEMES USED IN THE EXAM SYSTEM

The generation schemes used in the exam system were designed to be as powerful as possible within the above constraints. Generation schemes which use the information network approach and which stay within the constraints on pg/g's have not yet been developed in the exam system. Most pg/g's in the system use the random generation approach, assembly of pieces approach, or a combination of the two.

Figure 3.1 is a flowchart of the algorithm for the generation section of a problem generator/grader. Examples of some of the generation schemes used in the Generative Exam System are given below.

Figure 2.3 shows a typical problem generated by the pg/g on Fortran Expressions. Each expression tests one concept. The concepts that are tested in any given problem instance depend on

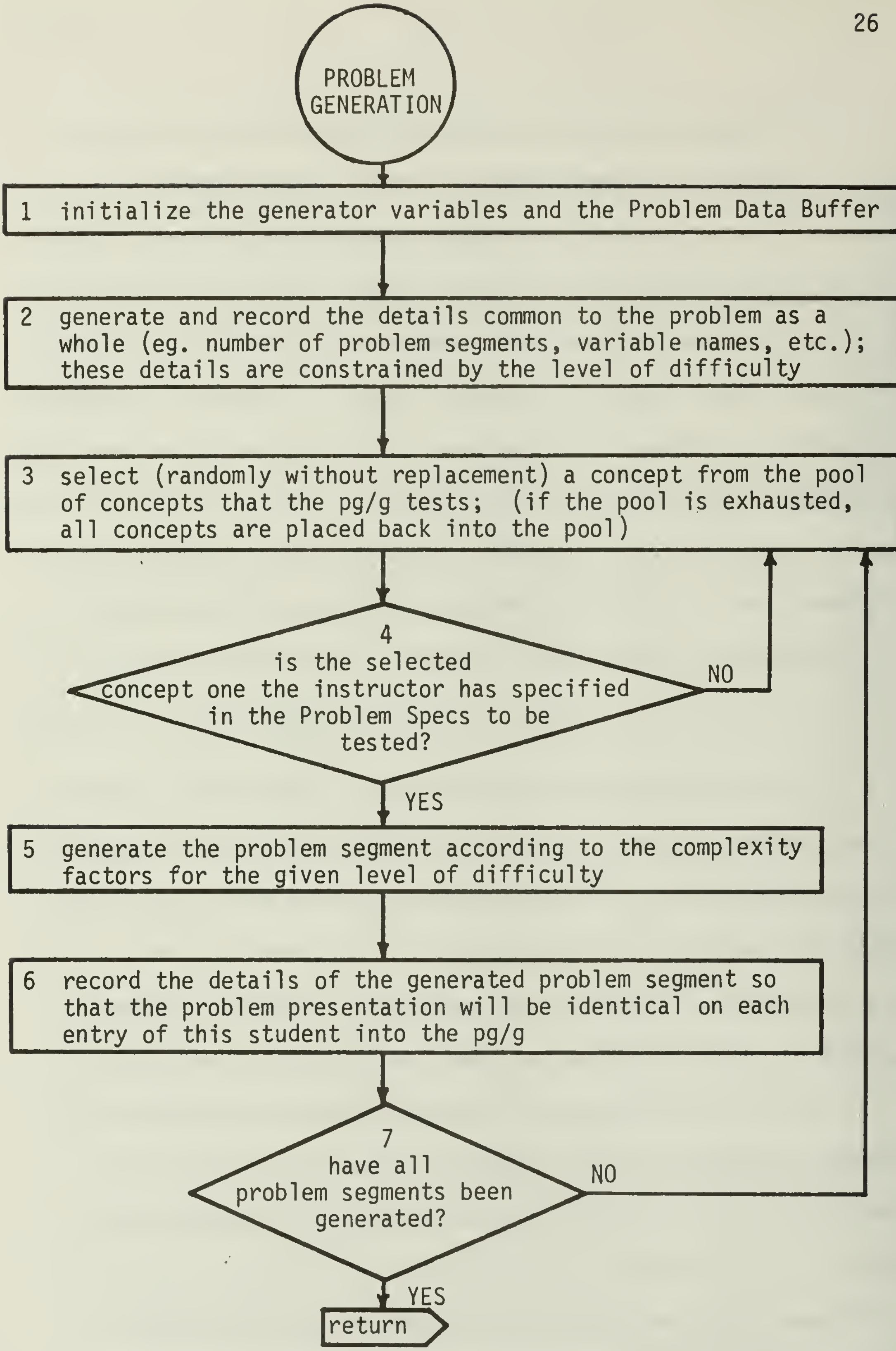


FIGURE 3.1: FLOWCHART OF THE ALGORITHM FOR THE GENERATION SECTION OF A PROBLEM GENERATOR/GRADER

which concepts the instructor has selected for testing and the level of difficulty of the problem. Table 3.1 shows which concepts may be tested for each level of difficulty. For a given difficulty, a concept is tested if there is an "X" for that concept under the difficulty level number and if that concept was selected by the instructor.

The level of difficulty is also used to determine the complexity of the problem. Table 3.2 lists the complexity factors for each level of difficulty.

The process used in generating an expressions problem is as follows. In this pg/g each problem segment consists of one expression. The names for the variables used in the problem are generated and stored in the Problem Parameters (block 2 in the flowchart). Then a concept is selected from the pool of concepts which the pg/g tests and which have been specified for testing by the instructor (blocks 3 and 4 in the flowchart).

An expression testing the selected concept is generated and recorded (blocks 5 and 6 in the flowchart) as follows. If "parentheses" is the selected concept, a parenthesis pattern is picked and placed in the appropriate positions in a buffer. The mode of the expression is then randomly selected unless it is determined by the concept selected. For example, if integer division is being tested, then the integer mode is used. Next the operators are selected and put in the appropriate positions in the buffer. Selecting operators may be constrained by certain operators that must be used (eg. a division when testing integer division) or that should not be used

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Concepts:										
precedence	X	X	X	X	X	X	X	X	X	X
parentheses		X	X	X	X	X	X	X	X	X
mixed-mode arithmetic			X	X	X	X	X	X	X	X
built-in functions					X	X	X	X	X	X
integer division					X	X	X	X	X	X
double exponentiation							X[1]	X[1]	X	X
unary minus							X[1]	X[1]	X	X

note 1: For difficulty levels 7 and 8, either double exponentiation or unary minus is tested, but not both in the same problem instance.

TABLE 3.1: CONCEPTS WHICH MAY BE TESTED IN A FORTRAN EXPRESSION PROBLEM FOR EACH LEVEL OF DIFFICULTY

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Complexity Factors:										
number of expressions:	2	3	4	5	6	6	7	7	8	8
number of operators:	2	2	2	3	3	3	3	4	4	4
number of constants:	3	3	2	2	1	1	1	0-1	0-1	0-1
number of variables	0	0	1	2	3	3	3	5-4	5-4	5-4
number of characters in variable names:	-	-	1	1	1	1-2	1-2	2-3	2-3	2-3
operators[1]:	A	A	A	A	A	B	B	B	B	B
pairs of parentheses:	-	1	1	1	1	2	2	2	2	3
magnitude of numbers[2]:										

note 1: Group A operators include + - * / . At least one of the operators in the expression must be / or * .
Group B operators include + - * / ** . At least one of the operators in the expression must be / or * . No consecutive exponents are allowed.
Expressions testing double exponentiation or unary minus are not constrained by this factor.

note 2: As the level of difficulty increases, the range of the numbers used in an expression increases.

TABLE 3.2: COMPLEXITY FACTORS USED FOR EACH LEVEL OF DIFFICULTY IN THE FORTRAN EXPRESSIONS PROBLEM GENERATOR/GRADER

(eg. two consecutive exponentiations if not testing double exponentiation). Following that, the operands are generated and placed in the appropriate positions in the buffer. Finally, the expression in the buffer is parsed and the correct answer is calculated. The results of all decisions made in the generation process are recorded in the Problem Parameters in the Problem Data Buffer so that the expression can be redisplayed each time the student returns to this problem.

More expressions are generated until the number appropriate to the level of difficulty has been produced.

Tables which drive the generation sections of other problem generator/graders are given in Appendix K.

A typical problem produced by the READ with FORMAT pg/g is shown in Figure 2.6. During generation, format concepts are selected from the pool of concepts chosen by the instructor when writing problem specifications. For each concept selected, appropriate format items are generated to compose the FORMAT statement in the problem. Corresponding values are generated for the data on the input card. The level of difficulty is used to guide the generation of the details of the problem. It further limits the pool of concepts that may be used in the problem, determines the magnitude of the numbers, sets the number of format items that appear in the problem, determines the size of the fields in the format items, influences where blanks may appear on the data card, and determines if there will be extra characters on the data card.

Figure 2.7 shows a typical problem generated by the pg/g on One-Dimensional Fortran Arrays. Generation of problems in this pg/g entails filling in the details in the structure of the program. The level of difficulty is used to determine the number of arrays, the number of elements in each array, whether the problem will contain an IF-loop, the complexity of the assignment statement, the array names, and the means used to initialize the arrays. The arrays may already be initialized when the student receives the problem or he may be required to show their initial contents as specified in the INTEGER statement or assignment statements in the program.

3.2 GRADING

Very little previous work has been done in the field of Computer Assisted Test Construction concerning the scoring of responses on exams except where the responses are totally correct or totally incorrect (eg. multiple choice, true/false, matching, and completion style questions). The techniques used by Barta (4) to grade program correctness and the theorem-proving techniques suggested by Goldberg (12) are too time consuming for use in the Generative Exam System. Since the Generative Exam System employs problems for which the solutions can be partially correct, grading schemes had to be developed which could equitably score partially correct solutions. The grading schemes used in the exam system are described below.

Responses in some of the pg/g's are selected from a list of possible responses (as in multiple choice questions). Scores in these pg/g's are determined by comparing the responses against the correct answers. This technique is widely used in other systems also.

In some pg/g's, such as the one on One-Dimensional Fortran Arrays, the student's responses are compared to the answers calculated by the pg/g during generation. This grading technique is similar to the preceding technique.

The pg/g's on Fortran Expressions and READ with FORMAT employ a partial credit grading scheme. In this scheme the response is checked for correct absolute value, correct sign, and correct mode. Partial credit is awarded for correct absolute value, correct absolute value and sign, or correct absolute value and mode. Full credit is awarded for a totally correct response. When writing problem specifications the instructor specifies the amount of credit to be awarded for a totally correct response and for each of the partially correct cases.

For example, if the correct answer for an expression is "-45.0", a response of "45" would be scored as correct absolute value, a response of "-45" would be scored as correct absolute value and sign, and a response of "45.0" would be scored as correct absolute value and mode.

A relative grading scheme is used by two other pg/g's. In the DO-loops Over an Expression pg/g, full credit is awarded if a response is absolutely correct (i.e. if it is the correct response for that position in the output in the completely correct answer), or if the response is correct relative to the previous response.

Figure 3.2 illustrates the scoring on a solution to a problem in the DO-loops Over an Expression pg/g. Scoring is weighted such that a correct value for the variable "W6" (the DO-loop indes) receives 2 points and a correct value for the variable "P6" receives 3 points. In the solution, the first value for "P6" is incorrect but if it is assumed that the first value for "P6" is correct, then the second value for "P6" is correct relative to the first value. The third value for "P6" is incorrect relatively and absolutely. (The relatively correct answer would be 79 and the absolutely correct answer is 77.) All of the remaining responses are correct relative to the third value of "P6". Incorrect responses are marked with three asterisks and relatively correct responses are marked with one asterisk.

In the PRINT with FORMAT pg/g, the accuracy and position of each response is checked. For each correct answer (value or character string printed) the closest matching response is located. Then the location of that response is compared to the correct position. Partial credit is awarded if the response is close in accuracy and/or position.

Figure 3.3 illustrates the scoring on a solution to a problem in the PRINT with FORMAT pg/g. Six items are checked in the response. Partial credit is awarded for the first item in the response ("5.2"). The decimal portion of the response is incorrect and the response is one column off in position. The second item ("6.61") is absolutely

.. Problem 5 DO-loop 3Ø points

REVIEW Total Score: 24 out of 3Ø

A score followed by *** means that answer is wrong.
A score for W6 followed by * means that answer is correct
assuming the previous W6 is correct.
A score for P6 followed by * means that answer is correct
assuming the corresponding W6 and previous P6 are right.

```
      INTEGER W6,P6,H2
      H2 = 29
      P6 = Ø
      DO  2Ø  W6 = 2, H2, 5
      IF (W6 .EQ. 12)  GOTO 2Ø
      P6 = P6 + 4 * W6 - 9
      PRINT, W6, P6
2Ø    CONTINUE
3Ø    CONTINUE
```

OUTPUT:

scores	W6	scores	P6
2.ØØ	2	Ø ***	1
2.ØØ	7	3.ØØ *	2Ø
2.ØØ	17	Ø ***	88
2.ØØ	22	3.ØØ *	167
2.ØØ	27	3.ØØ *	266
5.ØØ *	END of print		

CORRECT OUTPUT:

W6	P6
2	-1
7	18
17	77
22	156
27	255
END of print	

SHIFT-NEXT to next problem; SHIFT-BACK to previous problem
SHIFT-DATA to return to the cover page: 8

FIGURE 3.2: SCORING OF A SOLUTION TO A PROBLEM IN THE DO-LOOPS OVER AN EXPRESSION PROBLEM GENERATOR/GRADER

```
   Problem 3          F format PRINT                        30 points
_______________________________________________________________________

        REVIEW                       score:  25 out of 30

       A=5.28
       B=6.61
       C=8.5
       D=4.1
       E=1.646
       PRINT20,A,B,C,D,E
    20 FORMAT('1',2F6.2/' PCIB',2(F4.1,3X))

Your answer:
    1 column 10         20         30         40         50         60
line↓        ↓          ↓          ↓          ↓          ↓          ↓
 1   5.2  6.61
 2   PCIB 8.54.1
 3  1.6

Correct answer:
    1 column 10         20         30         40         50         60
line↓        ↓          ↓          ↓          ↓          ↓          ↓
 1    5.28  6.61
 2  PCIB 8.5    4.1
 3  1.6

_______________________________________________________________________
 SHIFT-NEXT to next problem;  SHIFT-BACK to previous problem
 SHIFT-DATA to return to the cover page;
```

FIGURE 3.3: SCORING OF A SOLUTION TO A PROBLEM IN THE PRINT WITH FORMAT PROBLEM GENERATOR/GRADER

correct in accuracy and relatively correct in position since there should be two blank columns between it and the first item. Similarly, the position of the third item ("PCIB") is incorrect but the position of the fourth item ("8.5") is relatively correct. The position of the fifth item ("4.1") is neither absolutely or relatively correct. The sixth item ("1.6") is totally correct.

In scoring, the total point value for the problem is ignored until the end of the process. Each item is assigned 5 points. One point is subtracted if the decimal portion of an item is incorrect. One point is deducted if the item is 1 or 2 columns off in position, 2 points deducted for 3 or 4 columns off, 3 points deducted for 5 to 10 columns off, and 4 points deducted for greater than 10 columns off. After each item is scored in this fashion, the points earned are weighted and a total percentage score is determined (see Table 3.3). This percentage score is multiplied by the total point value of the problem to arrive at the score earned by the student.

Item Answer	Maximum Points	Points Earned	% Points Earned	Item Weight	Weighted Percent
5.28	5	3	60%	.18	10.8%
6.61	5	5	100%	.18	18.0%
PCIB	5	4	80%	.10	8.0%
8.5	5	5	100%	.18	18.0%
4.1	5	3	60%	.18	10.8%
1.6	5	5	100%	.18	18.0%
					83.6%

83.6% of 30 points = 25 points

TABLE 3.3: SCORING OF THE PROBLEM ILLUSTRATED IN FIGURE 3.3

These relative grading schemes award credit for correct reasoning on problems where errors early in the problem solution have affected the later responses in the problem solution. The student may not lose the full value of the problem from his exam score because of an error made at the beginning of the problem.

Another approach to grading was used in the DO-loops Over an Array pg/g (written by Bert Speelpenning). The problems produced by this pg/g are quite similar to those produced by the DO-loops Over an Expression pg/g. Grading is done interactively. Each time the student enters a line of output, he is told if it is correct or not. If it is incorrect, some points are deducted from his score and he is given another chance. If his second attempt is also incorrect, more points are deducted from his score and he is shown the correct line of output and permitted to continue working. Thus errors committed early in the problem will not affect later responses.

While such interactive grading approaches were confusing when used in the same exam with traditionally graded problems (i.e. where the students were not told if their responses were right or wrong), interactive grading may be a valuable means of evaluating students and merits further research.

4. SYSTEM DEVELOPMENT

Several exams have been administered by the Generative Exam System during its development. Questionnaires were given after each use of the system to gather students' views of the exam system. Difficulties with the system encountered during these exams prompted several improvements to the system design.

4.1 EARLY EXAMS

The first exam administered by the system was given June 26, 1975. This was a practice exam given before the first hour exam in a small computer science class (CS 101 with about 40 students). The system worked well enough to demonstrate the feasibility of a generative exam system. The questions and responses from the questionnaire administered after the exam are summarized in Appendix B. Most students like the exam, perhaps because it was not difficult and did not count towards their grades in the course. About half of the students would have preferred having their next exam on PLATO.

The second exam administered by the system was given on July 31, 1975 to the same class as was the first exam. This exam was of average difficulty for an exam but was considerably more difficult than the first PLATO exam. It was part of the final exam and counted towards the students' grades in the course. A PLATO system failure caused the loss of data for some of the students who took the exam.

Responses to the questionnaire administered after the exam are summarized in Appendix C. Most students felt the instructions and procedures in the exam were hard to follow and most said they would prefer that their next exam be a written exam.

The third exam administered by the Generative Exam System was given on October 1, 1975. This exam was of average difficulty and counted as part of the students' grades in the course (CS 103 with about 75 students). During this exam, the auxilliary memory requirements for the exam system exceeded the amount allocated to the terminals used. This also caused the loss of data for some of the students who took the exam. Because of the loss of data that occurred in the second and third exams, the exam system was modified to store student data on disc as described in Section 2.1.2.

The questions and responses from the questionnaire administered after the third exam are summarized in Appendix D. Most students found the instructions difficult to follow and said they would prefer a written exam over the PLATO exam.

The fourth exam administered by the system was for an experiment concerned with the interactive aspects of the exam system. It is described in detail in the next section. The other two exams administered by the system were used in evaluations of the effectiveness of the exam system and of "tailored" exams. These are described in detail in Chapters 5 and 6.

Data and questionnaire responses from these last three exams indicate that as the Generative Exam System has been improved, students' reactions toward it have become more positive.

4.2 INTERACTIVE ASPECTS OF THE SYSTEM

The Generative Exam System was designed for a broad range of students. Since the majority of the students who would use the system would not be computer science majors and would not be very skilled in using a computer terminal, the dialogue in the exam system needed to be as "natural" as possible. It was also desirable to minimize the amount of typing required of students. This could be accomplished by requiring only short answers or selecting answers from a menu of possible answers (eg. multiple choice questions).

It was also considered desirable to minimize the distraction and confusion caused by taking an exam on PLATO as compared to taking a written exam. This was accomplished by making the PLATO exam look like a written exam, by allowing the student to return to each problem as often as he wanted, by redisplaying the same problem and the student's work each time he did return to a problem, and by permitting the student to change any of his answers without penalty during the course of taking the exam.

4.2.1 STUDENT-EXAM INTERACTION PROBLEMS

During the first three PLATO exams (see Section 4.1), it was noticed that students were spending about twice as long on their PLATO exams as would be expected if taking a similar written exam. To investigate this, an experiment was conducted in the fall of 1975 in which four subjects were videotaped while taking a short PLATO exam and a similar written exam. Their activities were classified and timed from the video tape. This experiment is described in detail in another document (9), but some of the results are described here.

The experimental subjects spent approximately twice as long on the PLATO exam as they did on the written exam (see Table 4.1). The subjects spent more time in the PLATO exam on thinking, on entering answers, and on exam management. Exam management included such activities as problem selection, waiting for the terminal to be loaded with special character sets, problem generation, problem presentation, and a category called "what next" which was the time subjects spent trying to find out how to go to the next problem, return to the cover page, etc.

Questionnaires administered during the experiment showed that the subjects felt the instructions were hard to follow but that typing ability and communicating with PLATO through the keyboard caused them little if any difficulty.

	PLATO exam	Written exam
average total Think time	13:18	9:30
average total time to Enter Answers	2:18	1:05
average total Exam Management time	6:18	:29
average total time	21:53	11:04

TABLE 4.1: AVERAGE TIMES SPENT ON TWO SIMILAR EXAMS, ONE ADMINISTERED ON PLATO, THE OTHER ON PAPER
Time is in minutes.

4.2.2 CORRECTIVE ACTIONS TAKEN

To decrease the amount of extra time students spent on PLATO exams, the system was modified in several ways.

Since most students worked through the problems in order, provisions were made to allow the student to move directly from one problem to the next without going to the cover page in between. Key conventions were adopted in all pg/g's so that pressing SHIFT-NEXT would take the student to the next "page" of his exam, SHIFT-BACK would take him to the previous "page", and SHIFT-DATA would take him to the cover page. Thus it became possible for the student to move through his exam without spending the time needed to display the cover page and type in the problem number of the next problem he wanted to work on.

The loading of special character sets was eliminated from all pg/g's. While this activity only took about twenty seconds each time it occurred, it was frustrating to sit idle while it was being done.

Originally, when the student first entered a problem, his problem was generated before anything was displayed on the screen. Again, it was frustrating to stare at a blank screen while the problem was being generated. To relieve this frustration, attempts were made to hide the time spent on generation. One way used was to display as much of the problem as possible before beginning generation so that the student would have something to read and think about while generation was going on. Further, if generating the whole problem took a long time, then parts of the problem could be displayed as they were generated. For example, in one of the pg/g's on Fortran Expressions, each expression is displayed as soon as it is generated so the student can begin to evaluate it before generation of the remaining expressions is completed.

To make the instructions and procedures in the exam easier to follow, similar tasks done in each problem were standardized among the pg/g's. For example, information identifying the problem is always displayed at the top of the screen and information about what to do next is always displayed at the bottom of the screen.

To make the screen as uncluttered as possible, pg/g authors are encouraged to carefully design the displays. Only the information that is actually needed to work the problem should be presented. Additional explanations can be given in HELP sequences. If a student enters an answer in a form unacceptable to the pg/g, then the pg/g should display a message explaining why the answer is unacceptable and what forms are acceptable. For example, in the pg/g on Fortran READ with FORMAT, the answers entered should be either real numbers or integers. If a response contains an "E" (scientific notation) it is not accepted and a message explains why it is not accepted and tells the student to enter an integer or a real number without an exponent.

The order in which material is presented on the screen can also help the student understand the problem. The order of display can lead the student through the problem in a logical sequence emphasizing tables and diagrams that the instructions refer to. Also, important details can be displayed first for emphasis.

To further make the student-exam interaction flow smoothly, the exam system uses the same key conventions throughout all parts of the system. These key conventions are also close to the key conventions used by PLATO and ACSES (20) so that a student's experience elsewhere on PLATO will not interfere with his taking of an exam in the Generative Exam System.

Changes to the exam system have eliminated much of the extra time spent on Exam Management in the PLATO exam and a little of the extra time spent on Thinking and Entering Answers. But the majority of the extra time spent on Thinking in the PLATO exam is still unexplained. Some hypotheses concerning this are offered here.

Working on PLATO was fairly new to most students. Further, taking an exam on PLATO was quite new to most of the students and the novelty of it all may have been more distracting than the students realized. Such a distraction could contribute to the additional Thinking time spent on the PLATO exam.

Students seem to hesitate when entering a response until they are reasonably sure that the response they enter is really the response they want. This behavior may be attributable to the fact that students do not realize they can change answers at any time without penalty, that they think it is difficult to change answers, or that they think the computer is going to let the number of previously entered responses to a question influence its grading of their final response to that question. This hesitation contributes to the extra time spent on Thinking and Entering Answers. As students become more familiar and comfortable with working on a computer terminal interactively and in particular with the exam system, this extra time should diminish.

Many students are distracted on a paper exam when the proctor looks over the student's shoulder at his work on the paper. This concern is more accentuated on the PLATO exam since the student's work is displayed on the screen which the proctors can easily see. This may also contribute to the hesitancy of students in entering answers since they spend more time rechecking answers before entering them.

Other factors that may contribute to the additional Think time and Entering Answers time on the PLATO exam include a lack of confidence in the computer or the programs to give the student full credit for his work; and resentment against having to work under the direction of a machine.

With the changes made to the Generative Exam System, thirty to forty percent of the extra time spent on the PLATO exam has been eliminated. Through the use of the Quiz System (1), which administers a short quiz at the end of each tutorial lesson, students could gain more familiarity and facility with taking exams on PLATO. This could lead to another thirty to forty percent reduction in the extra time students spent on PLATO exams. Any remaining extra time required to take an exam on PLATO may be acceptable when the advantages of using the Generative Exam System are considered.

5. COMPARISON OF PLATO EXAMS AND WRITTEN EXAMS

Two experiments have been conducted to evaluate the effectiveness of administering exams with the Generative Exam System. In each experiment a group of students took both a PLATO exam and a written exam. The data collected in these experiments was used to compare the effectiveness of PLATO exams with written exams. Data from the same experiments was used to evaluate the "tailored" style exam (see Chapter 6).

The first experiment was conducted on February 19, 1976. To control for some possible biases affecting the results of this experiment, a second experiment was conducted on July 6, 1976. These experiments are described below.

5.1 FEBRUARY EXPERIMENT

About 75 students from an introductory computer science course for business majors (CS 105) volunteered for the experiment. Each subject took a practice exam in the Generative Exam System four days before the class took their first written hour exam. The subjects were randomly assigned to take one of five different PLATO exams. Each PLATO exam contained the same three problems: one on Fortran expressions, one on DO-loops, and one on one-dimensional arrays. However, the problems differed in difficulty among the exams. In the "reg5" exam all problems were of difficulty level 5. In the "reg7" exam, all problems were of difficulty level 7 (which is more difficult than level 5), and the "reg10" exam contained problems of difficulty level 10 (the most difficult level). The "gambling" exam allowed each subject to select the difficulty level of his problems, and the "tailored" exam selected problem difficulty levels based on the subject's performance during the exam. These exams and the written exam are described in detail in Appendix E.

The experiment was conducted in the following fashion. After the subject signed onto the terminal, the exam system presented him with questions 1 and 2 on the questionnaire (the questionnaire and results are shown in Appendix F). The system then assigned each subject an exam based on his Student Record number. For example, every fifth subject received the tailored exam. An explanation of the procedures for the particular style of exam the subject was about to take was then displayed. When the subject had finished reading the explanation, his starting time was noted by the system and the exam was administered. Each subject was permitted to work on his exam for thirty minutes, but he could quit if he finished in less time. Upon completion of the exam, the subject was instructed to answer questions 3 and 4 on the questionnaire. The subject was then permitted to review the scores and answers on his exam. Finally, he answered questions 5, 6, 7, and 8 on the questionnaire and signed off the system. Four days later each subject took the written exam along with the rest of the CS 105 class.

The data collected during the experiment is listed in Appendix G.

Table 5.1 summarizes the results of the exams. It is assumed that the written exam was a valid measure of the subjects' knowledge. All of the PLATO exams except for the "reg5" exam showed good correlation with the written exam (.40 for the reg5 exam; .76, .60, .71, and .75 for the other exams). The results of the questionnaire showed that 70% of the subjects had spent less than 10 hours on PLATO before taking the PLATO exam, 88% of the subjects felt the instructions and procedures on the PLATO exam were clear or easy to follow, and 57% of the subjects would be willing to have at least part of their next exam on PLATO.

These questionnaire results indicate that the Generative Exam System had been developed to a point where students with relatively little experience on a computer terminal (i.e. with less than 10 hours of PLATO use) could take an exam at a terminal without feeling that the terminal interfered with their performance on the exam. Except for the reg5 exam group, the correlations between the PLATO exams and the written exam suggest that exams administered by the Generative Exam System are as effective at evaluating students as written exams.

These conclusions are clouded by the fact that administration of the PLATO exams and the administration of the written exam were four days apart and the fact that the PLATO exams were taken for practice (and thus did not count towards their grades in the course) and by volunteers from the course. The amount of time spent in preparation before the PLATO exams as compared to the time spent in

Subject Group:	reg5	reg7	reg10	gambling	tailored
Sample size:	18	19	19	18	16
Maximum possible score on the PLATO exam:	60	84	120	120	120
Mean score on the PLATO exam:	50.06	68.53	43.32	53.28	65.56
Standard deviation on the PLATO exam:	9.19	13.84	19.26	26.87	23.27
Maximum possible score on the written exam:	100	100	100	100	100
Mean score on the written exam:	50.33	62.95	54.47	52.61	51.50
Standard deviation on the written exam:	18.38	27.37	21.10	16.85	24.81
Correlation of PLATO total score to written total score:	.40*	.76*	.60*	.71*	.75*

TABLE 5.1: SUMMARY OF THE RESULTS FROM THE FEBRUARY EXPERIMENT
In the reg5 exam all problems were of difficulty level 5, in the reg7 exam all problems were of difficulty level 7, and in the reg10 exam all problems were of difficulty level 10. The gambling exam allowed each subject to select the difficulty level of his problems, and the tailored exam selected problem difficulty levels based on the subject's performance during the exam.
The * indicates that the correlation is significant at the .05 level.

preparation before the written exam could have varied greatly among the subjects. The motivation and attitudes of volunteers taking a practice exam could also be very different from those of students having to take an exam for a grade. These possible biases prompted another experiment.

5.2 JULY EXPERIMENT

The 75 students from an introductory computer science course for graduate students (CS 400) participated in this experiment. Each subject was required to take the PLATO exam and the written exam, and both counted towards their grades in the course. About half of the subjects took the PLATO exam the hour before the written exam, and the remaining subjects took the PLATO exam after the written exam.

The subjects were randomly assigned to take one of four PLATO exams. Each PLATO exam contained the same three problems: one on Fortran expressions, one on DO-loops, and one on Fortran READ with FORMAT. However, the problems differed in difficulty among the exams. In the "reg5" exam all problems were of difficulty level 5. In the "reg7" exam all problems were of difficulty level 7, and the "reg9" exam contained problems of difficulty level 9. The "tailored" exam selected problem difficulty levels based on the subject's performance during the exam. These exams and the written exam are described in detail in Appendix H.

The experiment was conducted as follows. At 10:00 a.m., about half of the subjects took the PLATO exams while the remaining subjects took the written exam. At 11:00 a.m., all subjects who had not taken the written exam at 10:00 took the written exam, and many of the subjects who had taken the written exam at 10:00 took the PLATO exams. Subjects who could not take the PLATO exams at 10:00 or 11:00 took them at 3:00 p.m. or at 7:00 p.m. After each subject had taken both exams, he was administered a questionnaire. The questions and results from the questionnaire are given in Appendix I.

The data collected during the experiment is listed in Appendix J. Table 5.2 summarizes the results of the exams. It is assumed that the written exam was a valid measure of the subjects' knowledge. The correlations between the PLATO exams and the written exam (.54, .45, .65, and .76) are not as high as found in the February experiment. This may be due to the fact that neither exam in the July experiment was comprehensive and thus may not have given full evaluations of subjects' knowledge of the course material.

The results of the questionnaire showed that 52% of the subjects had spent 10 or fewer hours on PLATO before taking the PLATO exam, 93% of the subjects felt the instructions and procedures on the PLATO exam were clear or easy to follow, and 57% of the subjects would be willing to have at least part of their next exam on PLATO. These results are very similar to the results obtained in the February experiment.

Table 5.3 shows the percentage of subjects who felt the exams were difficult or about right in difficulty. The similarity between the judged diffiulties of the written and PLATO exams and the fact

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	13	24	13	25
Maximum possible score on the PLATO exam:	50	70	90	100
Mean score on the PLATO exam:	41.77	55.08	57.31	56.20
Standard deviation on the PLATO exam:	5.29	13.48	24.33	18.88
Maximum possible score on the written exam:	100	100	100	100
Mean score on the written exam:	56.85	58.88	59.08	61.16
Standard deviation on the written exam:	26.68	24.68	21.60	22.69
Correlation of PLATO total score to written total score:	.54*	.45*	.65*	.76*

TABLE 5.2: SUMMARY OF THE RESULTS FROM THE JULY EXPERIMENT
In the reg5 exam all problems were of difficulty level 5, in the reg7 exam all problems were of difficulty level 7, and in the reg9 exam all problems were of difficulty level 9. The tailored exam selected problem difficulty levels based on the subject's performance during the exam.
The * indicates that the correlation is significant at the .05 level.

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	12	22	10	22
Subjects who felt the PLATO exam was difficult or about right in difficulty:	94%	68%	80%	81%
Subjects who felt the written exam was difficult or about right in difficulty:	84%	73%	90%	86%
Correlation of the judged difficulty of the PLATO exam to the judged difficulty of the written exam:	.67*	.64*	.75*	.47*

TABLE 5.3: RESULTS FROM THE JULY EXPERIMENT CONCERNING THE JUDGED DIFFICULTY OF THE EXAMS
The * indicates that the correlation is significant at the .05 level.

that subjects who felt the PLATO exam was relatively difficult also felt the written exam was relatively difficult (as indicated by the correlations (.67, .64, .75, and .47) between the judged difficulty of the PLATO exam and the judged difficulty of the written exam) suggests that the subjects viewed both the PLATO and the written exams as comparable in difficulty.

Table 5.4 shows the percentage of subjects who felt they showed a lot or all of their knowledge on the concepts tested in the exams. More subjects who took a regular style PLATO exam felt they were better able to demonstrate the extent of their knowledge on the PLATO exam (about 45%) than on the written exam (about 33%). The reverse situation in the tailored style PLATO exam may be

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	12	22	10	22
Subjects who felt they showed a lot or all of their knowledge of the concepts tested on the PLATO exam:	50%	45%	40%	18%
Subjects who felt they showed a lot or all of their knowledge of the concepts tested on the written exam:	33%	33%	30%	41%

TABLE 5.4: RESULTS FROM THE JULY EXPERIMENT CONCERNING THE PERCEIVED PERFORMANCE ON THE EXAMS
Perceived performance is how well a subject felt he was able to show the extent of his knowledge of the concepts tested.

attributable to the fact that subjects who took the tailored exam did not like it.

Table 5.5 shows the correlations concerning PLATO experience (i.e. the number of hours spent on PLATO). The fact that only one of the twelve correlations shown is significant at the .05 level suggests that PLATO experience does not provide an advantage in score or time spent on the PLATO exam.

Table 5.6 shows the times at which each exam was given, the number of subjects who took each exam at each time, and the mean score for each of these groups. Subjects who took the written exam at 10:00 took the PLATO exams at 11:00, 3:00, or 7:00. Subjects who took the written exam at 11:00, took the PLATO exams at 10:00. An analysis of variance showed that there is no significant

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	12	22	10	22
Correlation of PLATO experience to PLATO total score:	.53*	.19	-.41	.06
Correlation of PLATO experience to time spent on PLATO exam:	.34	-.24	.05	.18
Correlation of PLATO experience to the ease in understanding the instructions and procedures on the PLATO exam:	.11	.31	.39	.40

TABLE 5.5: CORRELATIONS CONCERNING PLATO EXPERIENCE FROM THE JULY EXPERIMENT
PLATO experience is the number of hours spent on PLATO before the experiment.
The * indicates that the correlation is significant at the .05 level.

	Time Taken	Subjects	Mean Score
Written Exam:	10:00 a.m.	45	57.18
	11:00 a.m.	30	62.53
PLATO Exams:	Time Taken	Subjects	Mean Score
	10:00 a.m.	30	51.3
	11:00 a.m.	24	51.1
	3:00 p.m.	16	56.5
	7:00 p.m.	5	69.0

TABLE 5.6: EXAM SCORES FOR THE SUBJECTS IN THE JULY EXPERIMENT GROUPED BY THE TIMES AT WHICH THEY TOOK THE EXAMS

difference in the means for the groups taking the written exam at different times (probability = .36). Similarly, an analysis of variance showed that there is no significant difference in the means for the groups taking the PLATO exams at different times (probability = .15). An analysis of covariance with the PLATO exam scores (grouped by time the exam was taken) as the experimental variable and the written exam scores as the covariate indicates that there is a significant difference between the mean scores for the PLATO groups (probability = .05). However, the assumption of homogeneity of regression in the analysis of covariance was not met, rendering this analysis questionable. These results suggest that the order in which the exams were taken had no significant effect on the scores earned. The effects of administering the PLATO exams at different hours during the day are still open to question.

The Generative Exam System gives slightly different questions even to students working at the same difficulty level. It has been suggested that this fact may cause some students to have more difficult exams than other students even though their exams are supposed to be equally difficult. The type and degree of variation among problems of the same difficulty can be predicted from the design of the generation schemes which are described in Chapter 3. The variations that can occur within a difficulty level are relatively small and should not significantly affect the difficulty of any given problem. Further, while there may be small differences in difficulty among questions generated at the same difficulty level, these differences would tend to average out over the entire exam for each student.

The results of the July experiment suggest that exams administered by the Generative Exam System are as effective at evaluating students as written exams, and that taking exams at the computer terminal does not hinder the students' performance.

6. THE TAILORED STYLE EXAM

A tailored exam attempts to find the level of each student's knowledge of the material being tested. As the student works his exam, problem difficulty levels are adjusted towards the student's knowledge level. If a student does well on a problem, he is given more difficult questions the next time he works on that problem.

A tailored exam is useful because it more accurately measures the extent of a student's knowledge. An accurate measurement of the extent of a student's knowledge in a subject area is the goal of domain-referenced testing and criterion-referenced grading with which a student is evaluated on his mastery of a set of concepts rather than on his score relative to the scores of other students. (For an example of a domain-referenced testing system, see Olympia (22).) Criterion-referenced grading of tests is often used in self-paced courses.

The tailored exam is similar to an oral exam in which the difficulty of the questions is increased or decreased depending on the degree of correctness of the student's responses to earlier questions.

A tailored exam should be less confusing and less frustrating to the student. The exam would be adapted to cover just the material he knew. This would reduce the confusion and frustration caused by guessing and working around concepts on the exam that the student did not know. Further, a tailored exam should be more efficient in terms of time. The exam would stop testing certain concepts if the student demonstrated sufficient knowledge of them and move on to testing other concepts.

On a broader level, a tailored exam would automatically administer an exam of a difficulty appropriate to the class. A single written exam which is too difficult or too simple for the class as a whole gives little information about the knowledge of individual students.

The design problems of implementing a tailored exam are discussed below. Then data from the experiments described in Chapter 5 is used to evaluate the effectiveness of tailoring an exam. This data indicates that the tailoring idea is effective but the current implementation of tailored exams in the Generative Exam System is inefficient in terms of time and is inpopular. Finally, a better approach to tailoring in the Generative Exam System is outlined.

6.1 IMPLEMENTING A TAILORED EXAM

In a tailored exam administered by the Generative Exam System, each time a student chooses to work on a problem, a difficulty level is assigned for the questions in that problem based on his previous work with the concepts tested in that problem. The maximum number of points a student may earn on a problem is proportional to the difficulty level of that problem. This section discusses the algorithm for determining that difficulty level.

The first consideration is the intial difficulty level of each problem. One approach is to use the same initial level for all students. Either an average level of difficulty or a high level of difficulty seem appropriate if this approach is adopted. However, it is obvious that tailoring would be more efficient if the intial level were closer to each student's level of knowledge. If the information were available to the tailoring algorithm, the grades that each student had earned in the course prior to the exam (such as on homework, quizzes, etc.) could be used to determine an initial level of difficulty for each problem in his exam. A student's grade point average would similarly be almost as useful to the tailoring algorithm. Another alternative is to ask each student to specify the level of difficulty at which he wants to start.

The current implementation of the tailored exam in the Generative Exam System uses a variant of this last alternative due to the unavailability of other scores for students or their grade point averages. At the beginning of a tailored exam, the student is asked what grade he expects to earn on the exam. From the response, the system calculates an initial level of difficulty for all problems on the exam.

The second consideration in a tailoring algorithm is the determination of the next difficulty level for a problem after it has been worked at least once. This next difficulty level could be a function of several things:

$$d_{k+1} = f(d_i, s_i, t_i, c_i, \ldots) \quad \text{for } i = 1 \text{ to } k$$

where d_i is the difficulty level of the ith problem entry,
s_i is the score earned on the ith problem entry,
t_i is the time spent in the ith problem entry,
c_i is the number of changes the student made to his responses in the ith problem entry,
etc.

The current implementation of the tailored exam uses:

$$d_{k+1} = f(d_k, s_k)$$

where d_k is the difficulty level of the kth problem entry, and
s_k is the score earned on the kth problem entry.

If the student earns greater than half of the points in a problem, then his level for that problem is increased in proportion to how well he did in the problem. For example, if a student earned 16 out of 20 points on a level 5 problem, then his level is raised to 8 for the next time he works that problem. Similarly, the student's level is decreased if he earns less than half the points on a problem.

A resistance to large changes in difficulty level is incorporated into the algorithm by limiting the amount of change in the level for a problem to no more than 3. For example, if a student earned 2 out of 20 points on a level 5 problem, his level would be reduced to 2 rather than 1.

The final consideration in a tailoring algorithm is determining which scores to keep for each problem. Ideally, the last difficulty level and score earned on a problem should be the best indication of the student's knowledge of the concepts in that problem. However, it can happen that a student will do well on a problem and return to a more difficult set of questions in that problem later. If he decides that his new set of questions is too difficult and leaves it unworked, and if he does not have time to return to that problem again later, then he will have a very low score and a high difficulty level for the last set of questions in that problem. To bypass this problem, the Generative Exam System keeps the highest score the student earns for each problem.

Several versions of the tailoring algorithm are evaluated in Section 6.2.3.

6.2 EVALUATION OF THE TAILORED EXAM

Data from the two experiments described in Chapter 5 has been used to evaluate the tailored style exam in the Generative Exam System. In drawing conclusions it is assumed that the written exams were valid measures of students' knowledge. The results are described below.

6.2.1 FEBRUARY EXPERIMENT

Table 6.1 shows the correlations between the PLATO exam scores and the written exam scores from the February experiment. In the tailored sample, only those subjects were included who had worked at least one problem more than once. Thus all subjects in the tailored

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	18	19	19	10
Correlation of PLATO exam total score to written exam total score:	.40*	.76*	.60*	.83*
Correlation of PLATO exam problem 1 score to written exam problem 2 score:	.002	.24	.48*	.79*
Correlation of PLATO exam problem 2 score to written exam problem 3 score:	.03	.47*	.41*	.73*

TABLE 6.1: CORRELATIONS OF PLATO EXAM SCORES AND WRITTEN EXAM SCORES FROM THE FEBRUARY EXPERIMENT The * indicates that the correlation is significant at the .05 level.

sample in Table 6.1 had experienced the effects of the tailoring algorithm at least once.

A strong correlation (.83) exists between the PLATO total score and the written total score for the tailored exam sample--stronger than for any other PLATO exam (.40, .76, and .60). Similarly, the correlations between the PLATO exam problems and similar problems on the written exam are stronger for the tailored sample (.79 and .73) than for any other PLATO exam.

These results suggest that the tailored exam is more effective at evaluating students than the regular style PLATO exams.

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	13	24	13	17
Correlation of PLATO exam total score to written exam total score:	.54*	.45*	.65*	.68*
Correlation of PLATO exam problem 1 score to written exam problem 1 score:	.36	.43*	.54*	.60*
Correlation of PLATO exam problem 2 score to written exam problem 2a score:	.33	.36*	.40	.28

TABLE 6.2: CORRELATIONS OF PLATO EXAM SCORES AND WRITTEN EXAM SCORES FROM THE JULY EXPERIMENT
The * indicates that the correlation is significant at the .05 level.

6.2.2 JULY EXPERIMENT

Table 6.2 shows the correlations between the PLATO exam scores and the written exam scores on the July experiment. As was done with the February experiment data, only those subjects were included in the tailored sample who had worked at least one problem more than once.

The results of this experiment show a strong correlation (.68) between the PLATO exam total score and the written exam total score for the tailored sample. This correlation is stronger than for the reg5 and reg7 PLATO exams (.54 and .45) but about the same as for the reg9 PLATO exam (.65).

The correlation of the first PLATO exam problem with the first written exam problem is strongest for the tailored sample (.60)--stronger than for any other PLATO exam. The low correlations between the second PLATO exam problem and the corresponding written exam problem may indicate that these two problems did not test the same concepts.

These results suggest that the tailored exam is at least as effective at evaluating students as the best regular style PLATO exam.

6.2.3 COMPARISON OF TAILORING ALGORITHMS

It has been suggested that the tailoring algorithm used in the Generative Exam System may bias comparisons of the tailored exam with the regular style PLATO exams. (Recall that the tailored exam in the Generative Exam System keeps the highest score earned on each problem.) To investigate this, other tailoring algorithms were studied.

Two independent tailoring schemes were tested in the February experiment (one was called "gambling" and the other was called "tailored"). In addition, modifications to the tailoring algorithms used in the July experiment could be studied from the data gathered. Five tailoring algorithms are described below.

Algorithm A: "tailored" exam

1. The initial level is set by what grade the student expected to earn on the exam.
2. The next difficulty level for a problem is based on the difficulty level and score earned on the previous entry into that problem.
3. The highest score earned on all sets of questions administered for a problem is the score kept for the problem.

Algorithm B: "gambling" exam

1. The initial level is selected by the student. The student selects how many points he wants to work for out of the total weight of the problem and a difficulty level is calculated from that.
2. The next difficulty level for a problem is selected by the student in the same way as is the initial difficulty level.
3. The student selects which set of questions he wants kept for each problem (without knowing the scores on any of them).

Algorithm C:

1. Same as algorithm A.
2. Same as algorithm A.
3. Keep the score on the last set of questions worked for a problem where it is not the case that the score was zero and the time spent was less the "t", where "t" is a small amount of time.

Algorithm D:

1. Same as algorithm A.
2. Same as algorithm A.
3. Keep the score for the set of questions from a problem for which the difficulty level was the last highest and on which the student earned 50% or more of the maximum possible score for that difficulty level. If no such case occurs, then keep the score as described in part 3 of algorithm C.

Algorithm E:

1. Same as algorithm A.
2. Same as algorithm A.
3. For each problem, take all sets of questions for a single level of difficulty on which the student earned a score within some small interval around the 50% score, average these scores, and keep this average as the score for the problem. If no such case occurs, then keep the score as described in part 3 of algorithm C.

Table 6.3 shows the sequence of problems worked by a subject taking a tailored exam in the July experiment. The scores kept for each algorithm (except algorithm B) are marked with an "X" under the column headed by the algorithm letter.

Table 6.4 shows the correlations of the PLATO exam total scores

Problem Number	Diff. Level	Score	Time Spent (minutes)	TAILORING ALGORITHM A	C[1]	D	E
1	9	16	10.7	X			
2	9	0	3.1				
2	6	24	6.2	X	X	X	X
1	10	13	4.9		X	X	X
3	9	0	1.0				
3	6	12	3.5				X[2]
3	6	12	3.6	X		X	X[2]
1	10	0	1.3				
3	6	6	1.5				X[2]
3	3	0	1.4				
3	1	4	0.5		X		
3	4	0	1.0				
			Total Score:	52	41	49	47

note 1: In algorithm C, the value for "t" is 1.5 minutes.

note 2: In algorithm E, these three scores are averaged together.

TABLE 6.3: SCORES EARNED USING TAILORING ALGORITHMS A, C, D, AND E FOR A SUBJECT FROM THE JULY EXPERIMENT

Tailoring Algorithm:	A	B	A	C	D	E
Experiment in which the algorithm was tested:	Feb.	Feb.	July	July	July	July
Sample size:	10	18	17	17	17	17
Correlation of the PLATO exam score using the specified tailoring algorithm to the written exam total score:	.83*	.71*	.68*	.64*	.63*	.58*

TABLE 6.4: CORRELATIONS OF PLATO EXAM TOTAL SCORE TO WRITTEN EXAM TOTAL SCORE FOR EACH TAILORING ALGORITHM
The * indicates that the correlation is significant at the .05 level.

to the written exam total scores for each tailoring algorithm. In the February experiment, algorithm A did better than algorithm B (correlations of .83 versus .71). In the July experiment, algorithm A did better than the other three algorithms but only slightly better (correlations of .68 versus .64, .63, and .58). It is concluded that the algorithm currently implemented in the Generative Exam System (algorithm A) does a slightly better job of tailoring than do the other algorithms studied.

6.2.4 STUDIES OF THE PROBLEM DIFFICULTY LEVELS

The tailored exam algorithm assumes that the distance between adjacent levels of difficulty is equal throughout the range. While insufficient data is available to test this assumption, the general relationship of one difficulty level to another in each problem generator/grader can be studied.

It is expected that good students would earn high scores on problems of all levels of difficulty, average students would earn high scores on low and middle levels of difficulty and lower scores on high levels of difficulty, and poorer students would earn high scores on low levels of difficulty and lower scores on higher levels of difficulty. Noting that the maximum number of points a student can earn on a problem is directly proportional to its difficulty level, these expectations are illustrated in the top three graphs in Figure 6.1.

To compare the actual performance to the expected performance, the subjects in each experiment were divided into three groups according to their scores on their written exam. Data for the two problems common to both exams was analysed. The mean score and number of subjects for each group, problem, and difficulty level are shown in Table 6.5.

The curves for problem 1 approximate the expected curves except for the curve for poor students which most closely resembles the expected curve for average students. This may not be surprising considering the fact that the concepts tested in this problem (expressions) are very basic and mastered by most students early in a course. The range of subjects tested in these experiments may have been a subset of the range the problem generator/grader is designed to test.

The curves for problem 2 approach the shape of the expected curves. Level 10 may be excessively difficult and level 5 may be a little too difficult.

As more data is collected, the difficulty level assignments in the pg/g's could be adjusted so that student performance curves approximate the expected curves.

6.2.5 ATTITUDES TOWARDS THE TAILORED EXAM

Table 6.6 shows a summary of the questionnaire results from the July experiment. The questionnaire and results are shown in Appendix I. In general, the results indicate that subjects who took the tailored exam did not like it.

Differences of about .5 between tailored subjects and other subjects on item 2 in the table suggest that tailored subjects

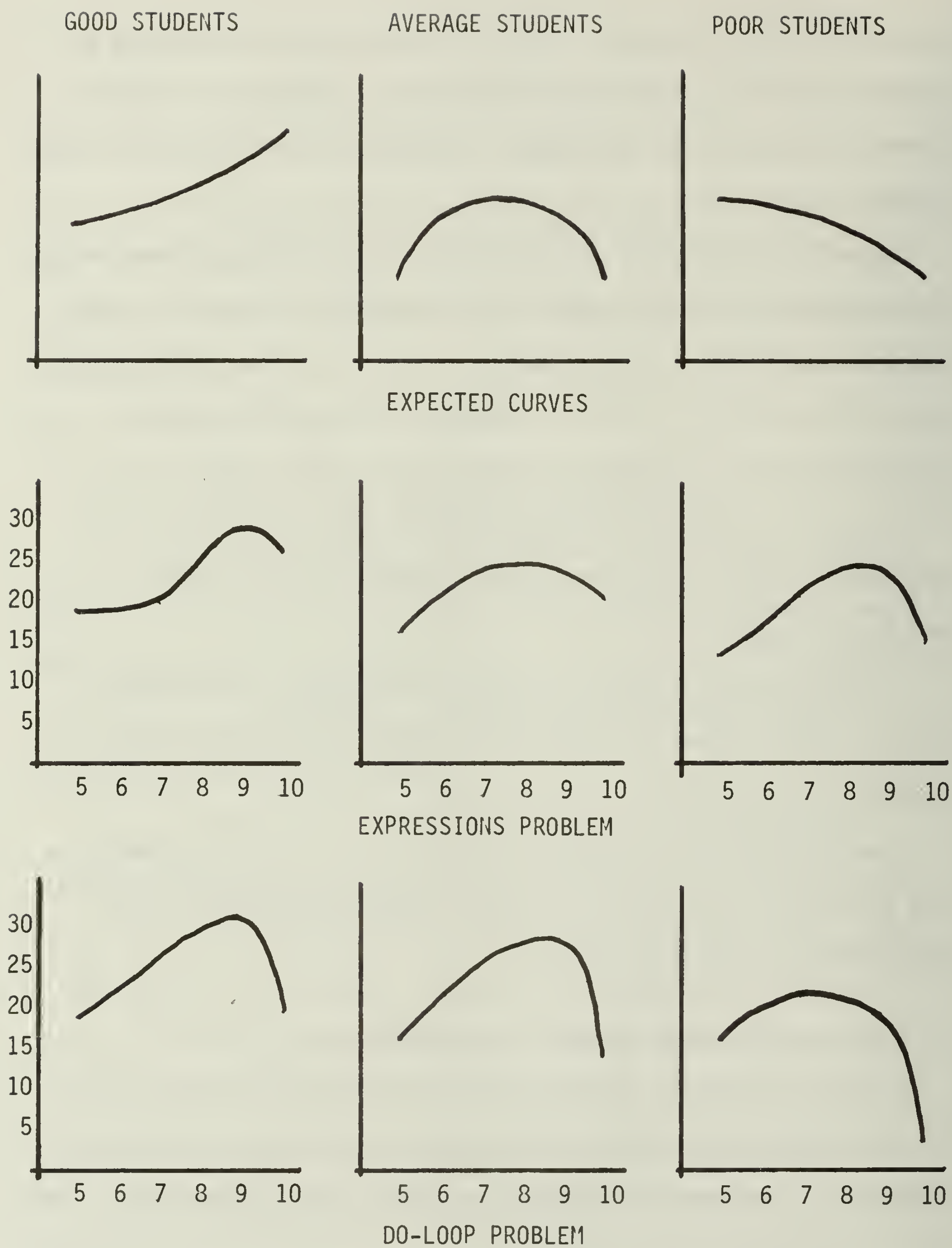


FIGURE 6.1: DIFFICULTY LEVEL VERSUS SCORE EARNED ON PROBLEMS FOR GOOD, AVERAGE, AND POOR STUDENTS FROM DATA COLLECTED IN THE FEBRUARY AND JULY EXPERIMENTS Difficulty level is plotted along the horizontal axis and score is plotted along the vertical axis.

GOOD STUDENTS				
Difficulty Level:	5	7	9	10
Expressions Problem				
Mean:	18.00	19.03	28.74	25.59
Sample size:	11	19	12	22
DO-loop Problem				
Mean:	18.36	25.55	30.57	19.06
Sample size:	11	20	7	16

AVERAGE STUDENTS				
Difficulty Level:	5	7	9	10
Expressions Problem				
Mean:	15.30	23.20	23.49	19.42
Sample size:	13	15	8	10
DO-loop Problem				
Mean:	16.23	24.80	27.75	13.63
Sample size:	13	15	8	8

POOR STUDENTS				
Difficulty Level:	5	7	9	10
Expressions Problem				
Mean:	12.64	19.47	23.10	13.60
Sample size:	11	19	9	5
DO-loop Problem				
Mean:	15.30	21.05	18.44	2.75
Sample size:	10	20	9	4

TABLE 6.5: MEAN SCORE AND NUMBER OF SUBJECTS FOR EACH GROUP OF STUDENTS, PROBLEM, AND DIFFICULTY LEVEL FROM DATA COLLECTED IN THE FEBRUARY AND JULY EXPERIMENTS Subjects in the February experiment are grouped as follows:

GOOD: written exam score was 59 or more,
AVERAGE: written exam score was 38 to 58,
POOR: written exam score was 37 or less.

Subjects in the July experiment are grouped as follows:

GOOD: written exam score was 74 or more,
AVERAGE: written exam score was 49 to 71,
POOR: written exam score was 46 or less.

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	12	22	10	22
2) Mean rating on ease of understanding the instructions and procedures (5 = very easy, 1 = very difficult):	3.83	3.77	3.90	3.36
3) Mean judged difficulty of PLATO exam (5 = very easy, 1 = very difficult):	2.83	3.18	2.80	2.27
4) Mean judged difficulty of written exam (5 = very easy, 1 = very difficult):	2.25	2.41	2.40	2.45
5) Mean rating on ability to show knowledge on the PLATO exam (4 = show all knowledge, 1 = show no knowledge):	2.50	2.23	2.40	1.77
6) Mean rating on ability to show knowledge on the written exam (4 = show all knowledge, 1 = show no knowledge):	2.33	2.05	2.30	2.18
7) Mean preference for a PLATO exam (3 = prefer PLATO, 1 = prefer written):	2.08	2.05	1.90	1.68
8) Mean preference for an individualized exam (2 = yes, 1 = no):	1.17	1.32	1.11	1.14

TABLE 6.6: SUMMARY OF THE QUESTIONNAIRE RESULTS FROM THE JULY EXPERIMENT

found the instructions more difficult to understand and the procedures more difficult to follow. Similar differences on item 3 indicate that the tailored subjects judged their PLATO exam as more difficult than the other PLATO subjects judged their exams. A difference of .5 or more exists between tailored and regular exam subjects in item 5 indicating that the tailored subjects felt that they were not able to show as much of their knowledge as the other PLATO exam subjects felt they were able to show. In item 7, tailored subjects showed a lower preference for PLATO exams than did regular exam subjects (1.68 versus 2.08, 2.05, and 1.90 on a 3 point scale). Item 8 suggests that all groups had strong preferences for regular exams over individualized exams.

From this data, it can be concluded that the tailored exam was unpopular.

6.2.6 EFFICIENCY OF THE TAILORED EXAM

Table 6.7 shows the data collected on the times spent in the PLATO exams in the July experiment. This data suggests that the tailored exam was inefficient in terms of time since subjects spent longer in it than in the other PLATO exams (an average of 41 minutes versus averages of 27, 32, and 40 minutes).

6.2.7 CONCLUSIONS

The results of the two experiments suggest that the tailored exam idea is effective at evaluating students but that the current implementation of the tailored exam in the Generative Exam System is inefficient in terms of time and unpopular with the students.

Subject Group:	reg5	reg7	reg9	tailored
Sample size:	13	24	13	25
Mean time spent on the PLATO exam:	26.92	31.38	39.46	40.32
Standard deviation:	8.44	8.37	8.96	7.40

TABLE 6.7: DATA ON THE TIMES SPENT ON THE PLATO EXAM IN THE JULY EXPERIMENT

6.3 SUGGESTIONS FOR IMPROVING THE IMPLEMENTATION OF THE TAILORED EXAM

The tailored exam, as currently implemented in the Generative Exam System, is inefficient in terms of time because a student must work a problem completely before any tailoring is done on the difficulty level for that problem. Since many problems in the exam system require several minutes to solve, working each of several problems two or three times requires a lot of time.

A solution to this problem is to handle tailoring independently in each problem generator/grader. The difficulty level could be adjusted after each question in a problem rather than after the complete set of questions in that problem. A student's knowledge of the concepts covered by a problem could be evaluated by working the problem once.

The general design for a tailoring pg/g could be as follows:

> Administer a question or two which test several concepts at a middle level of difficulty.

If the student does well, administer more difficult
questions each covering several concepts.

If the student does not do well, administer questions
covering fewer concepts or of lower difficult
or both.

The student leaves the problem if:
he has demonstrated adequate knowledge on all
concepts to be tested; or
he stabilizes at a level of difficulty that he
can handle but can not exceed; or
he decides to leave the problem. It is then assumed
that he was working at his level of knowledge
when he quit.

If the student returns to the problem, testing continues
at the level achieved before he left.

For example, in a problem generator/grader on Fortran expressions, a student would first be given an expression to solve testing precedence, parentheses, and mixed-mode arithmetic. If he solved it correctly, then he would be given an expression composed of more difficult constructs (such as integer division, double exponentiation, and unary minus). If he solved that correctly, the pg/g would inform the student that he had demonstrated sufficient knowledge in this area and should work on the other problems in the exam. If the student responded to the first expression incorrectly, then he would be given an expression which tested only precedence. If he got that wrong, he would receive another expression on precedence with a simpler sequence of operators. If the student solved this expression incorrectly also, the pg/g would move on and test other concepts individually (eg. parentheses alone). In this fashion, the pg/g would test each concept at a level of difficulty appropriate to the student.

A tailored exam utilizing pg/g's which tailor in this fashion could have the advantages of a tailored exam described at the beginning of this chapter. That is, it could more accurately evaluate the extent of each student's knowledge and do this in less time and with less frustration to the student than with conventional exams.

7. SUMMARY AND CONCLUSIONS

This paper has described the design, implementation, and evaluation of the Generative Exam System, a completely interactive system for the construction and administration of examinations. Since all tasks associated with examinations (from exam writing through analyses of exam results) are handled interactively in the system, the Generative Exam System offers many advantages over written exams. These advantages include a considerable savings in time and expense in writing, duplicating, and grading exams; exam security, provided by the fact that each student receives slightly different questions; consistent and accurate exam grading; the capability of allowing each student to review the scores and correct answers on his exam immediately after he finishes it; and the immediate availability of a complete analysis of exam results after a class finishes an exam.

The heart of this system is a set of problem generator/grader modules which produce examination problems. Generation and grading schemes used in the problem generator/graders were studied. The generation schemes produce a large number of similar problems by randomly generating numbers and character strings and assembling problem pieces into complete problem structures. The concepts covered by each problem and the level of complexity at which the concepts are tested may be altered under these generation schemes. The grading schemes award credit for partially correct responses by checking responses for variants of the correct answer or by grading the correctness of one response on the assumption that the previous response in that problem is correct.

Two experiments were conducted to evaluate the Generative Exam System. The coefficients for the PLATO exam scores correlated with the written exam scores averaged .64 in the February experiment and .60 in the July experiment suggesting that exams in the Generative Exam System are as effective at evaluating students as written exams.

The tailored style examination was then introduced. In a tailored exam, the difficulty levels of the problems are altered as the student works through the exam in an attempt to match the problem difficulty level to the student's level of knowledge. This approach should more accurately measure the extent of a student's knowledge and make this measurement in less time and with less frustration to the student than with the traditional style examination.

Data from the experiments conducted to evaluate the Generative Exam System was used to evaluate the tailored exam. The coefficients for the PLATO exam scores correlated with the written exam scores were higher for the group of students who took tailored exams than any other PLATO exam group (.83 versus .40, .76, and .60 in the February experiment, and .68 versus .54, .45, and .65 in the July experiment). These results indicate that the tailored exam idea is at least as effective in evaluating students as regular style exams. However, the implementation of the tailored exam in the Generative Exam System was inefficient in terms of time (tailored subjects spent an average of 40.32 minutes on their exam as opposed to an average of 31.78 minutes for the other subjects in the July experiment), and was unpopular (as indicated by the questionnaire responses). A new implementation for tailoring in the Generative Exam System was proposed which should make the tailored exam more efficient and less unpopular.

This study suggests that interactive exams are useful and effective in evaluating students and merit continued research, especially in the areas of problem generation and grading and tailored exams.

LIST OF REFERENCES

1. Anderson, Richard
The Quiz System
unpublished memo, Department of Computer Science, University of Illinois at Urbana-Champaign, Summer 1976

2. Ansfield, Paul J.
A User Oriented Computer Procedure for Compiling and Generating Examinations
Educational Technology, Vol. 13, No. 3, March 1973, p. 12-13

3. Baker, Frank B.
An Interactive Approach to Test Construction
Educational Technology, Vol. 13, No. 3, March 1973, p. 13-15

4. Barta, Ben Zion; and Whitlock, Lawrence R.
Documentation on ISAEP: Interactive System for Automatic Examination of Programming Skills
unpublished memo, Department of Computer Science, University of Illinois at Urbana-Champaign, February 1975

5. Brown, Willard A.
A Computer Examination Compositor for the IBM 360/40
Educational Technology, Vol. 13, No. 3, March 1973, p. 15-16

6. Buckley-Sharp, M. D.
A Multiple Choice Question Banking System
Educational Technology, Vol. 13, No. 3, March 1973, p. 16-18

7. Carbonell, J. R.
AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction
IEEE Transactions on Man-Machine Systems, Vol. MMM-11, No. 4, December 1970, p. 181-189

8. Denney, Cecil
There Is More to a Test Pool Than Data Collection
Educational Technology, Vol. 13, No. 3, March 1973, p. 19-20

9. Doring, Richard; Whitlock, Lawrence R.; and Hansen, Wilfred J.
An Evaluation of the Generative Exam System
Technical Report UIUCDCS-R-76-782, Department of Computer Science, University of Illinois at Urbana-Champaign, December 1975

10. Dudley, Thomas J.
How the Computer Assists in Pacing and Testing Student's Progress
Educational Tehcnology, Vol. 13, No. 3, March 1973, p. 21-22

11. Epstein, Marion G.
Computer Assisted Assembly of Tests at Educational Testing Service
Educational Technology, Vol. 13, No. 3, March 1973, p. 23-24

12. Goldberg,
CAI: The Application of Theorem-Proving to Adaptive Response Analysis
Stanford University, 1973

13. Hazlett, C. B.
MEDSIRCH: Multiple Choice Test Items
Educational Technology, Vol. 13, No. 3, March 1973, p. 24-26

14. Hsu, Tse-Chi; and Carlson, Marthena
Test Construction Aspects of the Computer Assisted Testing Model
Educational Technology, Vol. 13, No. 3, March 1973, p. 26-27

15. Jensen, Donald, D.
Toward Efficient, Effective, and Humaine Instruction in Large Classes: Student Scheduled Involvement in Films, Discussions, and Computer Generated Repeatable Tests
Educational Technology, Vol. 13, No. 3, March 1973, p. 28-29

16. Koffman, Elliot B.; Blount, Sumner; and Wei, Martin
CAI in Digital Logic Design, Debugging, and Programming
Computer and Electrical Engineering, Vol. 1, 1973, p. 299-320

17. Lippey, Gerald (ed.)
Computer Assisted Test Construction
Educational Technology Publications, Englewood Cliffs, New Jersey 07632, September 1974

18. McClain, Donald H.; Wessels, Stephen W.; and Sando, Kenneth M.
IPSIM - Additional System Enhancements Utilized in a Chemistry Application
Proceedings of the 1975 Conference on Computers in the Undergraduate Curricula, June 1975, p. 139-145

19. Meller, D. V.
Using PLATO IV
Computer-based Education Research Laboratory, University of Illinois, Urbana, Illinois, October 1975

20. Nievergelt, J.
Interactive Systems for Education -- The New Look of CAI
invited paper to the IFIP 2nd World Conference on Computer Education, Marseilles, France, September 1975

21. Olympia, P. L., Jr.
Computer Generation of Truly Repeatable Examinations
Educational Technology, Vol. 15, No. 6, June 1975, p. 53-55

22. Olympia, P. L., Jr.
Repetitive Domain-Referenced Testing Using Computers
Proceedings of the 1975 Conference on Computers in the Undergraduate Curricula, June 1975, p. 155-159

23. Paley, Roger M.
The Structure and Use of a Test Generating System Designed to Facilitate Individually Paced Instruction
On-Line, March 1976, p. 17-21

24. Prosser, Franklin
Repeatable Tests
Educational Technology, Vol. 13, No. 3, March 1973, p. 34-35

25. Ramani, S.; and Newell, A.
On the Generation of Problems
Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa. 15213, November 1973

26. Reynolds, Alan G.; and Flagg, Paul
Direct-Access Repeatable Testing in Statistics
Proceedings of Conference on Computers in the Undergraduate Curricula, June 1974, p. 211-214

27. Schonberger, Richard J.
Modular Instruction with Computer Assembled Repeatable Exams: Second Generation
Educational Technology, Vol. 15, No. 2, February 1975, p. 36-38

28, Seely, Oliver, Jr.; and Willis, Van
SOCRATES' Test Retrieval at the California State University and Colleges
Proceedings of the 1975 Conference on Computers in the Undergraduate Curricula, June 1975, p. 135-138

29. Smith, Stanley G.; and Sherwood, Bruce A.
Educational Uses of the PLATO Computer System
Science, Vol. 192, No. 4237, April 23, 1976, p. 344-352

30. Trocchi, Robert F.
Computer-Based Arithmetic Test Construction
Journal of Educational Data Processing, Vol. 10, No. 3, 1973

31. Uhr, Leonard
Teaching Machine Programs That Generate Problems as a Function of Interaction with Students
Proceedings of the 24th ACM National Conference, 1969, p. 125-134

32. Vickers, F. D.
Creative Test Generators
Educational Technology, Vol. 13, No. 3, March 1973, p. 43-44

33. Wexler, Jonathon D.
Information Networks in Generative CAI
IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, December 1970, p. 190-202

34. Whitlock, Lawrence R.
Documentation on the Generative Exam System
unpublished memo, Department of Computer Science, University of Illinois at Urbana-Champaign, June 22, 1976

APPENDIX A: PROBLEM GENERATOR/GRADERS AND AUTHORS

Listed below are the currently available problem generator/graders and their authors. Each pg/g's PLATO lesson name is enclosed in parentheses.

PROBLEM GENERATOR/GRADER	AUTHORS
Fortran expressions (csxfortexp)	Lawrence R. Whitlock
Fortran READ with FORMAT (csxfordfmt)	Lawrence R. Whitlock
Fortran PRINT with FORMAT (csxfoprfmt)	Lawrence R. Whitlock
DO-loops over an array (csxdoarray)	Bert Speelpenning
PL/1 IF-THEN-ELSE (csxif)	Lawrence R. Whitlock Wilfred J. Hansen Jurg Nievergelt
PL/1 syntax (csxpl1syn)	Francisco Izquierdo
Fortran syntax (csxfortsyn)	Mike Simons Francisco Izquierdo
Fortran DO-loops (csxpgg2)	Greg Peterson Fletcher Ross
Short answer questions (csxpgg3)	Tim Halvorsen Richard Doring
Fortran IF and GOTO statements (csxpgg5)	Woody Conrad
Fill-in-the-blank questions (csxpgg6)	Mitch Roth
DO-loops over an expression; with tailoring capabilities (csxdoexpr)	Lawrence R. Whitlock
One-dimensional Fortran arrays; with tailoring capabilities (csxshort)	Lawrence R. Whitlock
Fortran READ with FORMAT; with tailoring capabilities (csxpgg1)	Lawrence R. Whitlock
Fortran expressions; with tailoring capabilities (csxpgg4)	Lawrence R. Whitlock

APPENDIX B: QUESTIONNAIRE ADMINISTERED AFTER THE PLATO EXAM GIVEN JUNE 26, 1975

The following questionnaire was administered about a week after the exam was given to a CS 101 class taught by Prof. Murrell. Forty-one students completed the questionnaire. The number of students who selected each response is shown at the left of the response.

1. How did you like the PLATO exam compared to the written exam?

15 a. liked the PLATO exam much more than the written exam
3 b. liked the PLATO exam a little more than the written exam
4 c. liked the PLATO exam about the same as the written exam
3 d. liked the PLATO exam a little less than the written exam
11 e. liked the PLATO exam much less than the written exam

2. What did you think of the contents of the problem on Fortran expressions in the PLATO exam?

1 a. material tested was too difficult
7 b. material tested was challenging
22 c. material tested was of right difficulty
9 d. material tested was easy
2 e. material tested was too trivial

3. What did you think of the instructions and procedures for answering the questions in the problem on Fortran expressions in the PLATO exam?

6 a. very easy to follow
18 b. easy to follow
9 c. clear, but not obvious
4 d. difficult to follow
4 e. confusing

4. What did you think of the contents of the problem on Fortran READ and FORMAT statements in the PLATO exam?

4 a. material tested was too difficult
9 b. material tested was challenging
20 c. material tested was of right difficulty
4 d. material tested was easy
1 e. material tested was too trivial

5. What did you think of the instructions and procedures for answering the questions in the problem on Fortran READ and FORMAT statements in the PLATO exam?

5 a. very easy to follow
17 b. easy to follow
11 c. clear, but not obvious
5 d. difficult to follow
2 e. very confusing

6. What did you think of the contents of the problem on DO loops in the PLATO exam?

4 a. material tested was too difficult
12 b. material tested was challenging
16 c. material tested was of right difficulty
9 d. material tested was easy
1 e. material tested was too trivial

7. What did you think of the instructions and procedures for answering the questions in the problem on DO loops in the PLATO exam?

3 a. very easy to follow
8 b. easy to follow
16 c. clear, but not obvious
6 d. difficult to follow
9 e. very confusing

8. What did you think about grading in the PLATO exam?

3 a. grading was very easy
5 b. grading was easy
20 c. grading was about right
4 d. grading was hard
7 e. grading was very hard

9. What did you think about being able to review your PLATO exam immediately after completing it?

20 a. helped me learn the material in which I made errors
12 b. showed me what material I needed to study, but did not help me learn it
4 c. nice to know my grade, but it did not help me with the material
0 d. left me confused about the material tested
3 e. did not review my exam after completing it

10. Would you prefer that your next exam be on PLATO or be a paper and pencil exam?

16 a. PLATO exam
21 b. paper and pencil exam
3 c. don't care

11. How many times had you been on PLATO before you took the PLATO exam?

0 a. never before
8 b. once or twice before
16 c. three to five times before
3 d. six to ten times before
11 e. more than ten times before

12. Write any other comments you have on the PLATO exam.

APPENDIX C: QUESTIONNAIRE ADMINISTERED AFTER THE PLATO EXAM GIVEN JULY 31, 1975

The following questionnaire was administered on PLATO immediately after the exam was given to a CS 101 class taught by Prof. Murrell. The exam was part of the final exam for the course. Thirty-five students completed the questionnaire. The number of students who selected each response is shown at the left of the response.

1. How many times had you been on PLATO before you took this exam?
   - 0 a. never before
   - 1 b. once or twice before
   - 4 c. three to five times before
   - 10 d. six to ten times before
   - 20 e. more than ten times before

2. What did you think about taking an exam on PLATO?
   - 4 a. good environment for an exam
   - 12 b. satisfactory environment for an exam
   - 4 c. PLATO room is too noisy
   - 2 d. PLATO room is too crowded
   - 14 e. PLATO room is too crowded and noisy

3. What did you think of the content of this PLATO exam in general?
   - 6 a. material tested was too difficult
   - 12 b. material tested was challenging
   - 16 c. material tested was of right difficulty
   - 0 d. material tested was easy
   - 0 e. material tested was too trivial

4. What did you think of the instructions and procedures for getting around in the exam and answering questions?

0 a. very easy to follow
4 b. easy to follow
8 c. clear, but not obvious
13 d. difficult to follow
10 e. very confusing

5. What kind of an exam would you prefer?

2 a. exam on PLATO
20 b. paper and pencil exam
9 c. part of exam on PLATO and part on paper and pencil
4 d. don't care

6. Did you know that every student taking this exam worked slightly different questions?

22 a. yes
13 b. no

7. Given a set amount of time to work your exam, would you prefer

31 a. more easier questions
4 b. fewer more difficult questions

8. If your performance on the exam was monitored and evaluated while you worked, would you prefer

9 a. getting easier questions if you were not doing well. (Thus you could show what you know about the subject, but not get as many points for the questions as people who correctly answered the more difficult questions on the same subject.)
26 b. having all students receive questions of the same difficulty for each subject.

APPENDIX D: QUESTIONNAIRE ADMINISTERED AFTER THE PLATO EXAM GIVEN OCTOBER 1, 1975

The following questionnaire was administered about a week after the exam was given to a CS 109 class taught by Prof. Montanelli. Sixty students completed the questionnaire. The number of students who selected each response is shown under the response.

1. I preferred the PLATO exam to a written one covering the same material.

strongly agree	agree	neutral	disagree	strongly disagree
1	1	9	14	35

2. Rate the instructions and procedures for the 4 question types.

	very easy to follow	easy to follow	OK	hard to follow	very hard to follow
Arithmetic	13	20	18	5	3
Syntax	0	4	5	23	27
PRINT	0	1	12	27	16
READ	0	1	12	23	18

3. What did you think of the contents of each question?

	too difficult	difficult	OK	easy	too easy
Arithmetic	1	4	37	15	1
Syntax	9	23	22	4	0
PRINT	10	21	19	4	0
READ	14	24	13	3	0

4. What do you think of the following porperties of PLATO exams?

	worthwhile	neutral	worthless
Objective grading	21	25	13
Immediate grading	35	20	5
Ability to review	33	17	10
Different exams for everyone	18	19	23

5. What advantages did you see in the PLATO exam? What other advantages _might_ PLATO exams have (assuming that any faults and errors can be corrected)?

6. What disadvantages did you find in the PLATO exam? Were they specific to this exam, or would they pertain to any exam on PLATO?

APPENDIX E: DESCRIPTION OF THE EXAMS USED IN THE FEBRUARY EXPERIMENT

Five PLATO exams were used in the February experiment:

reg5: regular style exam of difficulty level 5
reg7: regular style exam of difficulty level 7
reg10: regular style exam of difficulty level 10
gambling style exam
tailored style exam

Each exam contained the same three problems, but of different difficulty levels. The problems covered the following material:

problem 1: Fortran expressions
problem 2: One-dimensional Fortran arrays
problem 3: Fortran DO-loops

Examples of these problems are given in Appendix L.

Figures E.1, E.2, and E.3 show the page of explanations associated with each PLATO exam style. Figure E.4 shows the cover page associated with the reg7 exam. The cover pages for the reg5 and reg10 exams are identical to the reg7 exam cover page except the total weight of the reg5 exam is 60 (20 points per problem) and the total weight of the reg10 exam is 120 (40 points per problem). Figure E.5 shows the cover page for the gambling exam, and Figure E.6 shows the cover page for the tailored exam.

Following Figure E.6, the written exam administered to the entire class is shown.

REGULAR STYLE EXAM EXPLANATION

When you are at the cover page, you may select any problem to work on.

When you are through working on a problem,
SHIFT-NEXT will take you to the next problem in the exam,
SHIFT-BACK will take you to the previous problem in the exam,
SHIFT-DATA will take you back to the cover page.

You may return to each problem as often as you want and your previous work will be there to modify.

You may look at this page anytime by pressing HELP while you are on the cover page.

Press NEXT to go to the cover page.

FIGURE E.1: PAGE OF EXPLANATIONS ASSOCIATED WITH THE REGULAR STYLE EXAM

GAMBLING STYLE EXAM EXPLANATION

When you are on the cover page, you may select any problem to work on. After selecting a problem, you will be asked to enter the number of points you want to work for. The more points you work for, the more difficult will be the questions in the problem; and the fewer points you work for, the easier will be the the questions in the problem. Thus, if you find the problem you get too difficult, return to the cover page and enter a different number of points to work for.

The second time you select to work on a problem, you will choose to get a new set of questions or to work more on the questions you had the previous time in that problem. You may work on each problem as often as you want.

After you have worked on a problem more than once you will choose which set of questions for that problem you want to have graded. Thus you can keep the questions you feel you did best on.

You may look at this page anytime by pressing HELP while you are on the cover page.

Press NEXT to go to the cover page.

FIGURE E.2: PAGE OF EXPLANATIONS ASSOCIATED WITH THE GAMBLING STYLE EXAM

TAILORED STYLE EXAM EXPLANATION

Each time you work on a problem in this exam, you will receive a new set of questions. Do your best to answer all the questions in that problem but do not spend an excessive amount of time. Once you leave a problem, you will not be able to work on those exact same questions again.

You should try to work through each problem at least two or three times. It is to your advantage to work each problem as many times as you can.

You may look at this page anytime by pressing HELP while you are on the cover page.

Press NEXT to go to the cover page.

FIGURE E.3: PAGE OF EXPLANATIONS ASSOCIATED WITH THE TAILORED STYLE EXAM

EXAM COVER PAGE (HELP for explanation)
CS1Ø5 experimental exam
Exam number 96, for course csa, for practice.
Maximum time allowed for this exam: 3Ø minutes.
Time you began: Ø2:11 Time now: Ø2:11 Time left: 3Ø min.

Number	Keyword	Weight	Score
1	Fortran expressions	28	
2	One dim. arrays	28	
3	Fortran DO-loop	28	
	TOTAL	84	

Select a problem: >
or Press SHIFT-LAB to quit and have your exam graded.
* (Asterisk means you have worked on this problem.)

FIGURE E.4: COVER PAGE ASSOCIATED WITH THE REGULAR STYLE EXAM OF DIFFICULTY LEVEL 7

GAMBLING EXAM COVER PAGE (HELP for explanation)
CS105 experimental exam
Exam number 90, for course csa, for practice.
Maximum time allowed for this exam: 30 minutes.
Time you began: 02:13 Time now: 02:13 Time left: 30 min.

num.	keyword	maximum point value	selected point value	score
1	Fortran expressions	40	0	
2	One dim. arrays	40	0	
3	Fortran DO-loop	40	0	
	TOTAL	120	0	

Select a problem: >
or Press SHIFT-LAB to quit and have your exam graded.
* (Asterisk means you have worked on this problem.)

FIGURE E.5: COVER PAGE ASSOCIATED WITH THE GAMBLING STYLE EXAM

TAILORED EXAM COVER PAGE (HELP for explanation)
CS105 experimental exam
Exam number 91, for course cs3, for practice.
Maximum time allowed for this exam: 30 minutes.
Time you began: 02:09 Time now: 02:09 Time left: 30 min.

Number	Keyword	Weight	Score
1	Fortran expressions	40	
2	One dim. arrays	40	
3	Fortran DO-loop	40	
	TOTAL	120	

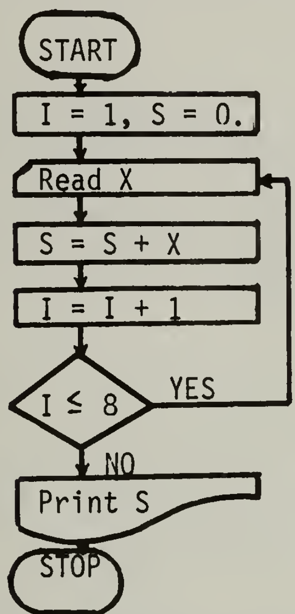
Select a problem: >
or Press SHIFT-LAB to quit and have your exam graded.
* (Asterisk means you have worked on this problem.)

FIGURE E.6: COVER PAGE ASSOCIATED WITH THE TAILORED STYLE EXAM

COMPUTER SCIENCE 105
HOUR EXAM 1
Feb. 23, 1976

Problem 1 (8 points)

(a) Convert the following flowchart to Fortran, by completing the partial program shown.




```
I = 1
S = 0.
```

(b) How many data cards are read?

Problem 2 If the following FORTRAN programs were executed, write below the values which would be printed.

(a) (8 points)

```
 1    INTEGER  I,COUNT
 2    I=0
 3    COUNT=1
 4    CONTINUE
 5    I=I+1
 6    IF(COUNT.GE.8) GO TO 9
 7    COUNT=COUNT+2
 8    GO TO 4
 9    PRINT,I,COUNT
10    STOP
      END
```

(b) (9 points)

```
1   I=2
2   J=3
3   K=4
4   A=4.0
5   B=1.5
6   C=0.5
7   X=B+J/I*I
8   M=(A+B)/(K*I)
9   S=4.0-C**(I/K)
10  PRINT,X,M,S
11  STOP
    END
```

Problem 3 (9 points)

Show the output of the following program, assuming that the data card has the following numbers:

5, 0, 8, 13, 3

```
1   INTEGER I,M(5)
2   READ,M
3   I=1
4   I=M(I)
5   PRINT,'I=',I
6   IF(I.LE.5) GO TO 4
7   STOP
    END
```

Problem 4 (8 points)

Assuming that the data cards are as shown below, give the output of the following program:

```
10  REAL A(4),B(4),R(4)
20  DO 70 I=1,4
30  READ,A(I),B(I)
40  R(I)=A(I)
50  IF(B(I).GT.A(I))  R(I)=B(I)
60  PRINT,R(I)
70  CONTINUE
80  STOP
    END
```

```
Card 1:   2, 14, 6.5
     2:   9, -2, 5.5
     3:   0, 10, 0.5
     4:  20, 30, -6.2
```

Problem 5 Each of the following WATFIV programs contains an error that will either prevent compilation or halt execution. Identify each error by the statement number, and describe the error briefly. Assume proper data is available for both programs.

(a) (10 points)

```
10  REAL X(10),Y(10)
20  READ,Y
30  DO 50 I-1,10
40  X(I)=2*Y(I)-Y(I+1)
50  PRINT,X(I)
60  STOP
    END
```

(b) (10 points)

```
10  REAL X(20),Y(20)
20  READ,X,Y
30  I=1
40  IF (I.GT.20)  STOP
50  IF (X.LE.0.)  GO TO 80
60  F=X(I)**Y(I)
70  PRINT,X(I),Y(I),F
80  I=I+1
90  GO TO 40
    END
```

Problem 6 Write WATFIV program segments that achieve each of the following:

(a) (12 points)
Read in a one-dimensional integer array X of 100 elements. Assign values to an integer array Y of the same length such that:

Y(I) = 0 if X(I) is odd
= 1 if X(I) is even

(b) (12 points)
Given 10 data cards with an integer M, 10≤M≤30, punched on each. Determine the smallest integer and print it.

Problem 7 (14 points)

A manufacturing firm had to lay off personnel to meet its budget. It was decided to lay off all personnel with 2 dependents or less, whose yearly salary exceeded $10,000. The FORTRAN program below is intended to determine the employees to be laid off, their number, and the total number of the employees before the lay-off. Complete the program by filling in the blanks. The last data card starts with a zero, and indicates the end of data.

```
C CS105 EXAM QUESTION
C
C   PROGRAM TO FIND EMPLOYEEES TO BE LAID OFF
C
C VARIABLE           DESCRIPTIONS
C   NAMES
C   SOCSEC           SOCIAL SECURITY NUMBER
C     NDEP           NUMBER OF DEPENDANTS
C     NEMP           NUMBER OF EMPLOYEES
C   LAYOFF           NUMBER OF EMPLOYEES TO BE LAID OFF
C   SALARY           SALARY FOR 1 EMPLOYEE
   10 INTEGER SOCSEC,NDEP,NEMP,LAYOFF
   20 REAL SALARY
   30 NEMP=0
   40 LAYOFF=__________
C BEGINNING OF THE INTERATION LOOP
   50 READ,SOCSEC,NDEP,SALARY
   60 IF(SOCSEC.EQ.0) GO TO __________
   70 __________
   80 IF(NDEP.GT.2) GO TO __________
   90 IF(SALARY.LE.10000) GO TO __________
  100 LAYOFF=LAYOFF+__________
  110 PRINT,'EMPLOYEE WITH SOCIAL SECURITY NUMBER = ',SOCSEC,
     *      'SHOULD BE LAID OFF'
  120 GO TO __________
  130 PRINT,'TOTAL NUMBER OF EMPLOYEES =',NEMP
  140 PRINT,'TOTAL NUMBER OF EMPLOYEES LAID OFF =',LAYOFF
  150 STOP
      END
```

APPENDIX F: QUESTIONNAIRE ADMINISTERED IN THE FEBRUARY EXPERIMENT

The following questionnaire was administered on PLATO immediately after the exams were given to volunteer subjects from a CS 105 class taught by Prob. Montanelli. Five different exams were given: regular style, difficulty level 5 (r5); regular style, difficulty level 7 (r7); regular style, difficulty level 10 (r10); gambling style (G); and tailored style (T). Eighty-five students completed the questionnaire. The number of students who selected each response is shown at the left of the response.

total	r5	r7	r10	G	T	
						1. How many times have you been on PLATO before you took this exam?
0	0	0	0	0	0	a. never before
0	0	0	0	0	0	b. once or twice before
25	3	5	9	7	1	c. three to five times before
35	9	8	5	8	5	d. six to ten times before
25	6	7	5	3	4	e. more than ten times before
						2. What grade do you expect to earn in CS 105 ?
26	6	7	8	3	2	a. A
47	10	11	9	10	7	b. B
10	2	2	1	4	1	c. C
1	0	0	1	0	0	d. D
0	0	0	0	0	0	e. F

total	r5	r7	r10	G	T	
						3. What did you think about the difficulty of this PLATO exam, on the average?
11	1	1	8	0	1	a. the questions were too difficult
35	3	5	7	14	6	b. the questions were challenging
32	12	9	4	4	3	c. the questions were of right difficulty
7	2	5	0	0	0	d. the questions were easy
0	0	0	0	0	0	e. the questions were too trivial
						4. Considering the difficulty of the questions you had on this exam, would you prefer
24	4	9	11			a. getting easier questions on material you did not know very well (but these questions would be worth fewer points) and getting harder questions (worth more points) on material you did know well so you could show the full extent of your knowledge.
33	14	11	8			b. having each student get the same difficulty questions as everyone else.
						4. Do you feel you could better show your knowledge by
				7		a. yourself selecting the difficulty of each question (by means of the number of points you want to work for).
				11		b. having each student get the same difficulty questions as everyone else.

total	r5	r7	r10	G	T	
						4. Would you prefer
					4	a. having just one set of questions in each problem which you could return to as often as you want (as on a written paper exam).
					6	b. having new questions each time so you have more opportunity of showing exactly what you know about each subject (as in the exam you just took).
						5. What did you think of the grading on this exam?
10	1	2	3	4	0	a. you deserved more credit
65	14	17	13	13	8	b. the exam was graded fairly
0	0	0	0	0	0	c. you deserved less credit
						6. What did you think of the instructions and procedures for getting around in the exam and answering questions?
17	5	6	2	3	1	a. very easy to follow
22	5	7	6	2	2	b. easy to follow
27	5	4	6	8	4	c. clear, but not obvious
7	0	2	1	3	1	d. difficult to follow
2	0	0	1	1	0	e. very confusing
						7. What did you think about taking an exam on PLATO?
14	6	2	2	4	0	a. good environment for an exam
44	7	13	8	9	7	b. satisfactory environment for an exam
6	0	2	3	1	0	c. PLATO room is too noisy
3	1	1	0	0	1	d. PLATO room is too crowded
8	1	1	3	3	0	e. PLATO room is too crowded and noisy
						8. What kind of an exam would you prefer?
19	7	5	4	2	1	a. exam on PLATO
32	3	10	7	9	3	b. paper and pencil exam
15	4	2	3	5	1	c. part of exam on PLATO and part on paper and pencil
9	1	2	2	1	3	d. don't care

APPENDIX G: DATA COLLECTED IN THE FEBRUARY EXPERIMENT

This appendix contains the data collected from the PLATO and written exams which was used in the analyses of the February experiment.

"Exam Group" refers to the PLATO exam style as follows:

Exam Group	PLATO Exam Style
1	Regular exam, difficulty 5
2	Regular exam, difficulty 7
3	Regular exam, difficulty 10
4	Gambling exam
5	Tailored exam

Table G.1 lists the means and standard deviations for the data collected for each PLATO exam group. Table G.2 shows the raw data. The sequence of problems worked by each subject who took the Gambling exam and the Tailored exam is shown in Tables G.3 and G.4.

Subject Group:	reg5	reg7	reg10	G	T	T
Sample Size:	18	19	19	18	10	16
Written Exam Scores						
total mean:	50.33	62.95	54.47	52.61	50.20	51.50
total std. dev.:	18.38	27.37	21.50	16.85	24.74	24.81
problem 2 mean:	9.56	11.21	8.68	10.89	9.30	7.25
problem 2 std. dev.:	5.79	5.58	6.98	3.58	7.23	7.39
problem 3 mean:	5.11	5.95	5.53	3.83	5.40	3.75
problem 3 std. dev.:	3.72	3.37	3.64	3.05	3.95	3.87
problem 4 mean:	5.67	5.68	5.47	5.22	5.00	3.62
problem 4 std. dev.:	3.16	3.15	3.12	3.37	3.56	3.81
PLATO Exam Scores:						
total mean:	50.06	68.53	43.32	53.28	67.90	65.56
total std. dev.:	9.19	13.84	19.26	26.87	25.35	23.27
problem 1 mean:	15.22	21.16	19.95	17.11	22.80	20.25
problem 1 std. dev.:	5.22	5.73	7.53	9.58	10.42	9.76
problem 2 mean:	19.44	24.47	11.11	20.33	23.90	24.94
problem 2 std. dev.:	1.76	6.96	11.53	12.36	13.35	11.38
problem 3 mean:	15.39	22.89	12.26	15.83	21.20	20.38
problem 3 std. dev.:	5.85	6.55	11.15	12.33	8.15	8.62
PLATO Exam Difficulty Levels						
problem 1 mean:	5	7	10	9.17	6.90	7.06
problem 1 std. dev.:	0	0	0	1.34	2.33	1.88
problem 2 mean:	5	7	10	8.28	6.60	6.88
problem 2 std. dev.:	0	0	0	2.08	3.41	2.70
problem 3 mean:	5	7	10	8.67	6.90	7.06
problem 3 std. dev.:	0	0	0	1.61	2.33	1.88

TABLE G.1: MEANS AND STANDARD DEVIATIONS FOR DATA COLLECTED IN THE FEBRUARY EXPERIMENT

EXAM GROUP	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P2	P3	P4
1	3	60	20	20	20	5.0	5	5	5	60	14	9	8
1	8	37	17	20	0	5.0	5	5	5	38	6	0	8
1	13	54	14	20	20	5.0	5	5	5	53	17	0	8
1	18	60	20	20	20	5.0	5	5	5	35	10	9	4
1	23	40	11	20	9	5.0	5	5	5	28	2	9	6
1	28	52	15	20	17	5.0	5	5	5	65	17	0	8
1	33	40	0	20	20	5.0	5	5	5	51	14	9	0
1	43	57	17	20	20	5.0	5	5	5	40	4	3	8
1	48	53	20	17	16	5.0	5	5	5	77	11	9	8
1	53	44	15	20	9	5.0	5	5	5	64	13	3	8
1	58	59	19	20	20	5.0	5	5	5	94	17	9	8
1	63	57	17	20	20	5.0	5	5	5	38	6	3	4
1	68	44	10	20	14	5.0	5	5	5	26	3	2	6
1	73	31	9	13	9	5.0	5	5	5	34	11	3	8
1	78	60	20	20	20	5.0	5	5	5	69	17	9	0
1	88	56	20	20	16	5.0	5	5	5	34	1	3	0
1	93	55	17	20	18	5.0	5	5	5	50	2	9	2
1	98	42	13	20	9	5.0	5	5	5	50	7	3	8

EXAM GROUP	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P2	P3	P4
2	4	80	24	28	28	7.0	7	7	7	77	17	9	8
2	9	82	28	28	26	7.0	7	7	7	100	17	9	8
2	14	66	10	28	28	7.0	7	7	7	70	14	9	8
2	19	80	24	28	28	7.0	7	7	7	100	17	9	8
2	24	38	5	21	12	7.0	7	7	7	22	6	3	0
2	34	67	20	28	19	7.0	7	7	7	94	17	9	8
2	39	76	20	28	28	7.0	7	7	7	77	14	9	6
2	44	84	28	28	28	7.0	7	7	7	89	11	9	8
2	49	78	22	28	28	7.0	7	7	7	79	10	9	8
2	54	66	19	26	21	7.0	7	7	7	28	7	2	0
2	59	78	24	28	26	7.0	7	7	7	55	14	0	6
2	64	82	26	28	28	7.0	7	7	7	55	3	3	8
2	69	78	22	28	28	7.0	7	7	7	97	17	9	8
2	74	49	19	21	9	7.0	7	7	7	29	7	3	2
2	79	75	26	28	21	7.0	7	7	7	77	17	9	0
2	84	60	20	14	26	7.0	7	7	7	34	5	3	6
2	89	66	18	27	21	7.0	7	7	7	33	2	3	2
2	94	50	22	19	9	7.0	7	7	7	46	15	3	8
2	99	47	25	1	21	7.0	7	7	7	34	3	3	6

TABLE G.2: RAW DATA COLLECTED IN THE FEBRUARY EXPERIMENT

EXAM GROUP	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P2	P3	P4
3	2	38	21	8	9	10.0	10	10	10	44	0	0	2
3	7	72	28	32	12	10.0	10	10	10	60	7	9	8
3	12	36	15	0	21	10.0	10	10	10	69	17	9	6
3	17	33	23	3	7	10.0	10	10	10	69	0	0	0
3	22	42	11	24	7	10.0	10	10	10	22	0	3	4
3	27	23	23	0	0	10.0	10	10	10	10	0	0	0
3	32	64	28	5	31	10.0	10	10	10	93	17	9	8
3	37	79	30	29	20	10.0	10	10	10	74	17	9	8
3	42	40	25	4	11	10.0	10	10	10	85	17	9	8
3	47	4	3	1	0	10.0	10	10	10	21	0	3	2
3	52	55	15	0	40	10.0	10	10	10	60	4	3	6
3	57	21	16	5	0	10.0	10	10	10	36	8	3	6
3	62	28	20	8	0	10.0	10	10	10	44	3	3	8
3	67	46	14	16	16	10.0	10	10	10	64	7	3	8
3	72	52	15	13	24	10.0	10	10	10	47	12	9	0
3	77	30	21	9	0	10.0	10	10	10	59	17	9	8
3	87	34	21	0	13	10.0	10	10	10	53	12	6	8
3	92	63	35	19	9	10.0	10	10	10	66	17	9	6
3	97	63	15	35	13	10.0	10	10	10	59	10	9	8

EXAM GROUP	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P2	P3	P4
4	5	105	30	35	40	10.0	10	10	10	59	13	9	0
4	10	20	0	20	0	10.0	10	10	10	40	7	3	8
4	15	72	33	12	27	8.7	10	8	8	59	11	3	8
4	20	62	27	30	5	8.0	8	8	8	49	11	3	2
4	25	50	20	3	27	10.0	10	10	10	52	11	3	0
4	30	34	18	10	6	6.7	10	5	5	43	7	3	8
4	35	20	3	10	7	8.0	8	8	8	29	6	3	4
4	45	48	17	16	15	8.0	8	8	8	47	14	3	8
4	50	45	18	16	11	8.0	8	8	8	42	7	3	6
4	55	75	20	40	15	10.0	10	10	10	85	9	9	8
4	60	48	16	22	10	8.7	9	8	9	48	17	0	8
4	65	103	35	32	36	10.0	10	10	10	82	14	9	4
4	70	52	11	20	21	7.7	9	5	9	67	14	3	8
4	75	33	13	11	9	9.3	10	10	8	63	17	3	8
4	80	18	9	0	9	4.3	5	3	5	24	8	0	0
4	85	91	15	40	36	10.0	10	10	10	57	7	0	6
4	90	33	15	14	4	9.3	10	8	10	32	9	3	0
4	95	50	8	35	7	10.0	10	10	10	69	14	9	8

TABLE G.2 (continued)

EXAM GROUP	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P2	P3	P4
5	6	87	36	24	27	9.3	9	9	10	61	17	9	8
5	21	78	21	31	26	7.7	7	9	7	55	17	9	8
5	41	90	28	36	26	7.7	7	9	7	51	14	9	0
5	46	66	21	23	22	7.0	7	7	7	42	6	0	6
5	51	62	13	28	21	7.0	7	7	7	59	2	3	6
5	61	33	12	4	17	3.7	3	1	7	35	6	0	8
5	66	49	22	1	26	5.0	7	1	7	20	0	3	6
5	86	101	34	36	31	9.0	9	9	9	77	17	9	0
5	91	87	35	40	12	9.0	10	10	7	92	14	9	8
5	96	26	6	16	4	2.7	3	4	1	10	0	3	0
5	11	74	24	28	22	7.0	7	7	7	63	17	3	8
5	16	62	9	28	25	7.0	7	7	7	37	6	3	0
5	26	75	22	28	25	7.0	7	7	7	39	0	0	0
5	31	70	14	28	28	7.0	7	7	7	87	0	0	0
5	36	69	21	36	12	9.0	9	9	9	79	0	0	0
5	56	20	6	12	2	7.0	7	7	7	17	0	0	0

TABLE G.2 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)	Set of Questions
5	1	10	35	6.3	new
	2	10	35	2.6	new
	3	10	40	3.3	new
	1	10	30	8.4	old
10	2	10	20	5.3	new
	3	10	0	2.0	new
	1	10	0	0.3	new
15	1	10	33	11.1	new
	2	10	0	2.5	new
	2	10	0	3.7	new
	2	8	12	7.4	new
	3	8	27	7.1	new
20	1	8	27	7.6	new
	2	8	30	6.8	new
	3	8	5	7.8	new
25	1	8	22	3.3	new
	2	8	30	4.8	new
	3	8	17	3.3	new
	1	10	20	4.4	new
	2	10	3	2.4	new
	3	10	27	3.2	new
30	1	6	7	2.5	new
	1	6	21	2.9	old
	2	6	13	3.9	new
	2	5	10	2.1	new
	3	5	6	2.4	new
	1	10	18	7.5	new
	2	8	10	3.7	new
	3	8	0	0.4	new
35	1	8	3	13.2	new
	1	10	0	0.5	new
	2	8	10	8.6	new
	3	8	7	5.2	new
	1	10	0	0.4	new

TABLE G.3: SEQUENCE OF PROBLEMS WORKED BY GAMBLING EXAM SUBJECTS

Subject Number	Problem Number	Diff. Level	Score	Time (min.)	Set of Questions
45	1	10	0	0.9	new
	1	5	17	3.4	new
	2	8	16	5.0	new
	3	5	13	2.5	new
	1	8	17	4.5	new
	2	10	0	1.5	new
	3	8	15	3.5	new
	3	10	0	0.7	new
50	1	8	18	8.6	new
	2	10	0	0.4	new
	2	8	16	6.7	new
	3	10	3	3.2	new
	3	8	11	5.5	new
55	1	10	20	10.1	new
	2	10	40	7.1	new
	3	10	15	7.5	new
60	1	9	16	5.0	new
	2	8	22	3.6	new
	3	8	0	1.3	new
	3	5	12	1.5	new
	3	7	27	2.2	new
	2	9	18	3.9	new
	2	9	17	2.5	new
	1	9	16	1.8	old
	3	9	10	1.8	new
	2	9	18	2.7	new
65	1	10	35	10.6	new
	2	10	32	10.9	new
	3	10	36	4.7	new
70	1	6	0	2.0	new
	1	5	3	1.0	new
	2	5	20	3.0	new
	2	9	0	1.6	new
	3	5	20	3.7	new
	3	9	21	5.0	new
	1	9	11	6.3	new

TABLE G.3 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)	Set of Questions
75	1	10	13	9.1	new
	2	10	11	10.4	new
	3	8	9	4.2	new
80	1	5	9	10.3	new
	2	5	7	3.4	new
	2	3	0	2.5	new
	3	5	2	4.0	new
	3	5	9	4.1	new
	2	3	0	0.2	old
	1	5	9	0.2	old
85	1	10	15	13.4	new
	2	10	40	8.3	new
	2	10	40	0.2	old
	3	10	36	11.1	new
90	1	8	0	1.3	new
	1	10	15	5.9	new
	2	10	0	2.8	new
	2	8	14	3.9	new
	3	10	4	3.2	new
95	1	10	8	13.5	new
	2	10	35	7.6	new
	3	10	7	5.5	new

TABLE G.3 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
6	1	9	36	7.7
	2	9	24	4.9
	3	9	5	5.8
	3	3	12	1.0
	3	10	0	0.3
	2	10	19	2.8
	3	1	4	0.5
	2	9	0	0.2
	3	10	27	3.0
	1	10	20	2.4
21	2	7	23	1.7
	3	7	26	4.4
	3	10	21	3.6
	1	7	21	6.2
	2	9	31	3.3
	3	10	15	3.6
	1	9	19	3.8
	2	10	0	0.2
	3	8	0	0.2
	2	1	4	0.8
	3	1	3	0.6
41	1	7	28	6.2
	2	7	19	5.4
	3	7	26	5.1
	1	10	25	6.1
	2	9	36	2.6
	3	10	11	3.2
46	1	7	21	7.2
	2	7	23	5.3
	3	7	22	3.5
	1	9	14	4.9
	2	9	18	2.9
	3	9	12	2.3
51	1	7	13	4.3
	2	7	28	3.0
	3	7	21	3.7
	1	7	0	0.3
	2	10	0	0.1

TABLE G.4: SEQUENCE OF PROBLEMS WORKED BY TAILORED EXAM SUBJECTS

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
61	1	7	6	9.9
	2	7	0	0.2
	1	3	12	1.3
	1	10	0	0.5
	2	1	4	1.4
	3	7	17	3.5
	1	1	4	0.4
	2	10	0	4.0
	3	8	5	4.3
	1	10	5	3.9
66	1	7	0	4.5
	2	7	0	0.3
	1	1	4	1.0
	2	1	1	1.4
	3	7	26	4.0
	1	10	13	6.6
	2	1	1	0.6
	3	10	11	3.4
	1	7	22	3.7
	2	1	1	0.4
	3	6	18	1.8
86	1	9	34	13.7
	2	9	36	3.3
	3	9	31	7.1
	1	10	25	8.8
91	1	7	12	4.2
	2	7	28	3.7
	3	7	12	3.2
	1	6	24	3.7
	2	10	37	4.2
	3	6	11	1.5
	1	10	35	6.6
	2	10	40	3.0
96	1	6	4	6.6
	2	6	8	3.1
	3	6	0	10.0
	1	3	6	2.2
	2	4	16	1.9
	3	1	4	1.9

TABLE G.4 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
11	1	7	24	9.0
	2	7	28	6.1
	3	7	22	5.5
16	1	7	9	6.0
	2	7	28	5.6
	3	7	25	5.1
26	1	7	22	4.3
	2	7	28	3.2
	3	7	25	4.4
	1	9	0	0.7
31	1	7	14	5.1
	2	7	28	1.9
	3	7	28	3.3
36	1	9	21	10.1
	2	9	36	3.9
	3	9	12	8.8
56	1	7	6	12.3
	2	7	12	16.3
	3	7	2	9.2

TABLE G.4 (continued)

APPENDIX H: DESCRIPTION OF THE EXAMS USED IN THE JULY EXPERIMENT

Four PLATO exams were used in the July experiment:

reg5: regular style exam of difficulty level 5
reg7: regular style exam of difficulty level 7
reg9: regular style exam of difficulty level 9
tailored style exam

Each exam contained the same three problems, but of different difficulty levels. The problems covered the following material:

problem 1: Fortran expressions
problem 2: Fortran DO-loops
problem 3: Fortran READ with FORMAT

Examples of these problems are given in Appendix L.

Figure H.1 and H.2 show the page of explanations associated with each PLATO exam style. Figure H.3 shows the cover page associated with the reg7 exam. The cover pages for the reg5 and reg9 exams are identical to the reg7 exam cover page except the total weight of the reg5 exam is 50 (10 points for problem 1 and 20 points each for problems 2 and 3) and the total weight of the reg9 exam is 90 (18 points for problem 1 and 36 points each for problems 2 and 3). Figure H.4 shows the cover page for the tailored exam.

Following Figure H.4, the written exam administered in the experiment is shown.

REGULAR STYLE EXAM EXPLANATION

When you are at the cover page, you may select any problem to work on.

When you are through working on a problem,

SHIFT-NEXT will take you to the next problem in the exam,

SHIFT-BACK will take you to the previous problem in the exam,

SHIFT-DATA will take you back to the cover page.

You may return to each problem as often as you want and your previous work will be there to modify.

You may look at this page anytime by pressing HELP while you are on the cover page.

Press NEXT to go to the cover page.

FIGURE H.1: PAGE OF EXPLANATIONS ASSOCIATED WITH THE REGULAR STYLE EXAM

TAILORED STYLE EXAM EXPLANATION

This exam contains 3 problems. But each time you work on a problem, you will receive a new set of questions. Thus if you work on each problem 3 times, you will have worked 9 sets of questions (3 sets for each problem).

You should do your best on each set of questions but do not spend an excessive amount of time. Once you leave a problem, you will not be able to work on that set of questions again.

You should try to work through each problem at least two or three times. It is to your advantage to work each problem as many times as you can.

You may look at this page anytime by pressing HELP while you are on the cover page.

To insure that you understand the directions, tell me how many sets of questions you will have worked if you work problem 1, then problem 2, then problem 1 again. >

FIGURE H.2: PAGE OF EXPLANATIONS ASSOCIATED WITH THE TAILORED STYLE EXAM

EXAM COVER PAGE (HELP for explanation)
CS400 PLATO Hour Exam 1 (7)
Exam number 121, for course csa, for a grade.
Maximum time allowed for this exam: 40 minutes.
Time you began: 09:06 Time now: 09:06 Time left: 40 min.

Number	Keyword	Weight	Score
1	Expressions	14	
2	DO Loops	28	
3	Formatted READ	28	
	TOTAL	70	

Select a problem: >
or Press SHIFT-LAB to quit and have your exam graded.
* (Asterisk means you have worked on this problem.)

FIGURE H.3: COVER PAGE ASSOCIATED WITH THE REGULAR STYLE EXAM OF DIFFICULTY LEVEL 7

```
TAILORED EXAM COVER PAGE                          HELP for explanation
CS400 PLATO Hour Exam 1   (T)
Exam number 123, for course csa, for a grade.
    Maximum time allowed for this exam:  40 minutes.
    Time you began: 09:08  Time now: 09:08  Time left: 40 min.
```

Number	Keyword	Weight	Score
1	Expressions	20	
2	DO Loops	40	
3	Formatted READ	40	
	TOTAL	100	

```
Select a problem:   >
or  Press SHIFT-LAB to quit and have your exam graded.
*   (Asterisk means you have worked on this problem.)
```

FIGURE H.4: COVER PAGE ASSOCIATED WITH THE TAILORED STYLE EXAM

COMPUTER SCIENCE 400
MIDTERM
July 6, 1976

Problem 1 FORTRAN EXPRESSIONS & ASSIGNMENTS (30 points)

For each of the following FORTRAN assignment statements indicate:
a) the type (REAL, INTEGER, or MIXED) of the expression on the right hand side of the equal sign,
b) the value of the expression on the right hand side of the equal sign, and
c) the value of the variable on the left hand side of the equal sign after execution of the statement.

Assume default types for variables and the following initial values:

I = 3 J = 2 B = 2. A = 3.

```
1)  C = (A*A + B*B)**1/2
2)  K = A*B + 1/2*I
3)  L = B**I**J
4)  M = 2*I/5*5
5)  D = 3*J**2
6)  N = I/J - I*J - 2.3
```

Problem 2 LOOPS

For each of the following program segments, indicate on the lines provided what is printed by the program segment. Do not worry about format or left to right spacing on the line. You need only have the correct values in the correct order on the correct line.

(a) (18 points)

```
      I = 1
      DO 10 J = 2,5
      DO 20 K = J,I
      PRINT,I,J,K
   20 CONTINUE
      I = 2*I
   10 CONTINUE
```

(b) (12 points)

```
      N = 0
      NS = 0
      I = 1
   20 IF(I/2*2.EQ.I) GO TO 10
      N = N + 1
      NS = NS + I
      PRINT,N,NS
   10 I = I + 1
      IF(I.LE.8) GO TO 20
```

<u>Problem 3</u> PROGRAMMING (40 points)

Write a complete FORTRAN program that:
1) reads the value N from a card (you may use FORMAT-free input),
2) calculates the value of A, B, and A/B, where:

$$A = \sum_{i=1}^{N} \left[(i^3 - N)^2 \right] \qquad B = \sum_{i=1}^{N} \left[(N^3 - i)^{3/2} \right] \qquad \text{and}$$

3) prints the value of N, A, B, and A/B appropriately labeled.

You may assume $N > 0$. If you wish, you may use the space below to make a flowchart. However, it will not be used for grading purposes. Start your program on the next page. This problem can be programmed in less than 10 statements. You will not receive full credit if you use more than 15 statements.

APPENDIX I: QUESTIONNAIRE ADMINISTERED IN THE JULY EXPERIMENT

The following questionnaire was administered to each subject in the July experiment after he had taken both the PLATO and written portions of the CS 400 midterm exam. Four different PLATO exams were given: regular style, difficulty level 5 (r5); regular style, difficulty level 7 (r7); regular style, difficulty level 9 (r9); and tailored style (T). Sixty-six subjects completed the questionnaire. The number of students who selected each response is shown at the left of the response. For each question, the weight of each response is shown to the right of the letter naming that response.

total	r5	r7	r9	T	
					1. How many hours have you spent on PLATO before this exam (for other courses and projects as well as for CS 400)?
					2. Without regard for question content, rate the clarity of the instructions and the procedures for entering answers and moving from question to question in the PLATO portion of the exam.
9	3	4	1	1	a. 5 very easy to follow
31	4	10	7	10	b. 4 easy to follow
21	5	7	2	7	c. 3 clear but not obvious
5	0	1	0	4	d. 2 difficult to follow
0	0	0	0	0	e. 1 very difficult to follow

total	r5	r7	r9	T	
					3. Rate the general level of difficulty of the <u>PLATO portion</u> of the exam.
2	0	2	0	0	a. 5 PLATO portion was trivially easy
8	1	5	1	1	b. 4 PLATO portion was easy
34	8	10	7	9	c. 3 PLATO portion was about right in difficulty
17	3	5	1	8	d. 2 PLATO portion was difficult
4	0	0	1	3	e. 1 PLATO portion was very difficult
					4. Rate the general level of difficulty of the <u>written portion</u> of the exam.
0	0	0	0	0	a. 5 written portion was trivially easy
3	0	2	0	1	b. 4 written portion was easy
29	5	9	5	10	c. 3 written portion was about right in difficulty
25	5	7	4	9	d. 2 written portion was difficult
9	2	4	1	2	e. 1 written portion was very difficult
					5. Rate how you feel you performed on the <u>PLATO portion</u> of the exam.
18	1	7	1	9	a. 1 I was not able to show what I knew about the concepts tested
24	5	5	5	9	b. 2 I was able to show a little of what I knew about the concepts tested
20	5	8	3	4	c. 3 I was able to show a lot of what I knew about the concepts tested
4	1	2	1	0	d. 4 I was able to show all of what I knew about the concepts tested
					6. Rate how you feel you performed on the <u>written portion</u> of the exam.
10	1	4	0	5	a. 1 I was not able to show what I knew about the concepts tested
32	7	10	7	8	b. 2 I was able to show a little of what I knew about the concepts tested
22	3	7	3	9	c. 3 I was able to show a lot of what I knew about the concepts tested
1	1	0	0	0	d. 4 I was able to show all of what I knew about the concepts tested

total	r5	r7	r9	T	
					7. Which would you prefer?
15	4	7	1	3	a. 3 next exam be entirely on PLATO
21	3	6	2	10	b. 1 next exam be entirely written
30	5	9	7	9	c. 2 next exam be part written and part on PLATO
					d. 2 don't care
					8. What kind of an exam would you prefer?
12	2	6	1	3	a. 2 an <u>individualized</u> exam--getting more difficult questions worth more points on concepts I knew well and easier questions worth fewer points on concepts I did not know well
50	10	13	8	19	b. 1 an exam where all students receive questions of the same difficulty and point value
					9. If you took the PLATO quizzes after working the PLATO lessons in this course, did that experience make it easier to take part of the midterm on PLATO?
42	8	18	7	9	a. 2 yes
22	2	4	3	13	b. 1 no
0	0	0	0	0	c. 0 I did not take any of the PLATO lesson quizzes

APPENDIX J: DATA COLLECTED IN THE JULY EXPERIMENT

This appendix contains the data collected from the PLATO and written exams which was used in the analyses of the July experiment.

"Exam Group" refers to the PLATO exam style as follows:

Exam Group	PLATO Exam Style
1	Regular exam, difficulty 5
2	Regular exam, difficulty 7
3	Regular exam, difficulty 9
4	Tailored exam

Table J.1 lists the means and standard deviations for the data collected for each PLATO exam group. Table J.2 shows the raw data. The sequence of problems worked by each subject who took the Tailored exam is shown in Table J.3.

Subject Group:	reg5	reg7	reg9	T	T
Sample Size:	13	24	13	25	17
Written Exam Scores					
total mean:	56.85	58.88	59.08	61.16	62.94
total std. dev.:	26.68	24.68	21.60	22.69	22.82
problem 1 mean:	18.69	20.71	20.08	20.80	20.53
problem 1 std. dev.:	6.73	5.43	6.09	5.98	5.54
problem 2a mean:	8.08	7.54	8.77	9.40	9.82
problem 2a std. dev.:	7.59	7.11	7.34	6.76	6.98
problem 2b mean:	4.69	6.42	5.54	5.64	5.82
problem 2b std. dev.:	4.77	5.40	3.95	4.74	4.75
problem 3 mean:	25.38	24.21	24.69	25.32	26.76
problem 3 std. dev.:	12.76	12.28	11.81	12.77	12.14
PLATO Exam Scores					
total mean:	41.77	55.08	57.31	56.20	56.35
total std. dev.:	5.29	13.48	24.33	18.88	21.36
problem 1 mean:	8.23	11.17	13.00	12.88	12.88
problem 1 std. dev.:	1.92	2.99	3.46	4.34	4.78
problem 2 mean:	19.54	25.50	22.15	27.56	27.41
problem 2 std. dev.:	1.20	4.41	12.77	8.84	9.64
problem 3 mean:	14.00	18.42	22.15	15.76	16.06
problem 3 std. dev.:	4.69	8.99	12.47	9.88	11.08
PLATO Exam Difficulty Levels					
total mean:	5	7	9	8.17	8.13
total std. dev.:	0	0	0	1.21	1.31
problem 1 mean:	5	7	9	8.52	8.65
problem 1 std. dev.:	0	0	0	1.48	1.66

TABLE J.1: MEANS AND STANDARD DEVIATIONS FOR DATA COLLECTED IN THE JULY EXPERIMENT

Subject Group:	reg5	reg7	reg9	T	T
Sample Size:	13	24	13	25	17
PLATO Exam Difficulty Levels					
problem 2 mean:	5	7	9	8.28	8.29
problem 2 std. dev.:	0	0	0	1.43	1.61
problem 3 mean:	5	7	9	7.72	7.47
problem 3 std. dev.:	0	0	0	1.65	1.84
PLATO Exam Times					
total mean:	26.92	31.38	39.46	40.32	41.47
total std. dev.:	8.44	8.37	8.96	7.40	5.59
problem 1 mean:	10.12	10.44	12.86	10.20	88.88
problem 1 std. dev.:	3.58	3.35	3.02	4.41	3.98
problem 2 mean:	5.35	7.98	9.58	7.24	6.54
problem 2 std. dev.:	1.60	3.16	3.12	4.29	4.50
problem 3 mean:	11.31	12.69	17.04	10.48	7.68
problem 3 std. dev.:	5.46	5.60	6.56	7.68	6.41

TABLE J.1 (continued)

GROUP EXAM	TIME	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO TIMES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P1	P2A	P2B	P3
1	1	5	47	7	20	20	33.0	10.8	5.1	17.6	5.0	5	5	5	57	24	0	0	33
1	1	11	40	10	20	10	38.0	13.8	4.9	18.4	5.0	5	5	5	52	17	4	5	26
1	1	17	35	8	20	7	19.0	9.0	5.4	4.9	5.0	5	5	5	57	20	18	2	17
1	1	23	30	3	20	7	36.0	9.6	7.1	19.4	5.0	5	5	5	25	9	5	3	8
1	1	29	43	10	20	13	31.0	8.5	5.0	16.9	5.0	5	5	5	86	22	16	12	36
1	2	35	43	10	20	13	22.0	11.6	4.0	6.2	5.0	5	5	5	53	19	2	0	32
1	2	41	41	8	20	13	12.0	4.6	2.9	4.7	5.0	5	5	5	84	28	16	3	37
1	2	47	40	7	20	13	27.0	11.9	8.1	6.6	5.0	5	5	5	6	6	0	0	0
1	2	53	44	8	16	20	36.0	17.3	6.8	12.4	5.0	5	5	5	35	21	0	4	10
1	3	59	50	10	20	20	33.0	13.9	7.5	10.9	5.0	5	5	5	86	22	17	12	35
1	3	65	40	9	18	13	27.0	7.6	4.7	13.9	5.0	5	5	5	60	14	6	8	32
1	3	71	42	9	20	13	20.0	7.0	4.7	8.5	5.0	5	5	5	40	13	3	0	24
1	4	77	48	8	20	20	16.0	5.9	3.3	6.6	5.0	5	5	5	98	28	18	12	40

TABLE J.2: RAW DATA COLLECTED IN THE JULY EXPERIMENT

GROUP EXAM	GROUP TIME	STUD. NUM.	PLATO SCORES TOTAL	P1	P2	P3	PLATO TIMES TOTAL	P1	P2	P3	PLATO DIFF. TOTAL	P1	P2	P3	WRITTEN SCORES TOTAL	P1	P2A	P2B	P3
2	1	3	65	12	25	28	25.0	8.3	8.9	7.6	7.0	7	7	7	68	24	0	9	35
2	1	6	37	5	21	11	39.0	11.4	12.0	16.2	7.0	7	7	7	28	19	2	2	5
2	1	9	67	14	28	25	27.0	11.4	6.0	9.9	7.0	7	7	7	82	20	16	12	34
2	1	12	57	8	28	21	34.0	15.2	7.4	10.7	7.0	7	7	7	54	17	2	12	23
2	1	15	68	12	28	28	29.0	10.5	6.4	9.9	7.0	7	7	7	95	29	18	12	36
2	1	18	66	13	28	25	28.0	9.8	6.8	11.8	7.0	7	7	7	40	13	5	0	22
2	1	21	42	14	28	0	24.0	6.1	5.5	12.6	7.0	7	7	7	71	21	16	0	34
2	1	24	59	12	26	21	38.0	10.0	13.2	15.3	7.0	7	7	7	54	26	2	1	25
2	1	27	36	10	26	0	34.0	11.9	6.8	15.6	7.0	7	7	7	49	21	3	1	24
2	1	30	59	9	25	25	40.0	9.8	5.1	24.4	7.0	7	7	7	74	19	4	12	39
2	2	33	56	14	28	14	17.0	6.1	3.5	7.0	7.0	7	7	7	87	20	16	12	39
2	2	36	62	10	27	25	22.0	5.4	8.0	8.0	7.0	7	7	7	21	15	2	2	2
2	2	39	62	13	28	21	25.0	9.9	5.9	9.0	7.0	7	7	7	86	24	18	12	32
2	2	42	10	3	7	0	29.0	12.3	13.3	2.7	7.0	7	7	7	15	9	0	0	6
2	2	45	51	14	26	11	45.0	10.6	6.9	27.7	7.0	7	7	7	85	28	18	4	35
2	2	48	53	13	22	18	25.0	7.3	6.3	11.9	7.0	7	7	7	49	20	5	0	24
2	2	51	68	14	26	28	37.0	13.2	7.8	15.5	7.0	7	7	7	35	19	0	12	4
2	2	54	51	11	26	14	23.0	5.7	5.0	11.7	7.0	7	7	7	54	19	0	12	23
2	3	57	49	9	26	14	41.0	17.9	8.2	14.6	7.0	7	7	7	24	14	0	0	10
2	3	60	55	11	23	21	24.0	7.6	7.2	9.0	7.0	7	7	7	86	30	14	11	31
2	3	63	59	8	26	25	46.0	16.3	10.7	18.6	7.0	7	7	7	46	24	9	6	7
2	3	66	70	14	28	28	22.0	8.4	4.0	8.4	7.0	7	7	7	100	30	18	12	40
2	3	69	53	11	28	14	35.0	14.6	10.5	9.0	7.0	7	7	7	64	20	9	10	25
2	4	75	67	14	28	25	44.0	10.9	16.1	17.5	7.0	7	7	7	46	16	4	0	26

TABLE J.2 (continued)

GROUP		STUD.	PLATO SCORES				PLATO TIMES				PLATO DIFF.				WRITTEN SCORES				
EXAM	TIME	NUM.	TOTAL	P1	P2	P3	TOTAL	P1	P2	P3	TOTAL	P1	P2	P3	TOTAL	P1	P2A	P2B	P3
3	1	2	23	16	0	7	44.0	10.1	9.1	24.1	9.0	9	9	9	44	16	6	3	19
3	1	8	22	9	13	0	38.0	12.4	11.3	14.8	9.0	9	9	9	32	15	9	0	8
3	1	14	43	12	6	25	50.0	14.4	8.5	27.2	9.0	9	9	9	34	8	0	6	20
3	1	20	90	18	36	36	21.0	11.9	5.9	4.0	9.0	9	9	9	87	28	18	6	35
3	3	26	86	16	34	36	40.0	19.6	13.2	7.2	9.0	9	9	9	81	28	6	12	35
3	1	32	43	15	6	22	38.0	12.4	6.5	18.3	9.0	9	9	9	75	23	16	5	31
3	2	38	36	13	16	7	29.0	11.9	5.5	11.7	9.0	9	9	9	28	18	0	0	10
3	2	44	63	6	32	25	57.0	17.9	16.3	23.0	9.0	9	9	9	62	15	18	2	27
3	2	50	71	16	30	25	36.0	11.6	8.0	16.7	9.0	9	9	9	70	28	0	6	36
3	2	56	45	11	23	11	46.0	11.0	12.7	21.7	9.0	9	9	9	67	23	6	10	28
3	3	62	86	14	36	36	38.0	10.2	8.2	19.1	9.0	9	9	9	88	22	18	12	36
3	3	68	53	9	22	22	40.0	14.4	9.1	16.9	9.0	9	9	9	63	21	2	6	34
3	4	74	84	14	34	36	36.0	9.4	10.3	16.8	9.0	9	9	9	37	16	15	4	2

TABLE J.2 (continued)

GROUP EXAM	GROUP TIME	STUD. NUM.	PLATO SCORES TOTAL	PLATO SCORES P1	PLATO SCORES P2	PLATO SCORES P3	PLATO TIMES TOTAL	PLATO TIMES P1	PLATO TIMES P2	PLATO TIMES P3	PLATO DIFF. TOTAL	PLATO DIFF. P1	PLATO DIFF. P2	PLATO DIFF. P3	WRITTEN SCORES TOTAL	WRITTEN SCORES P1	WRITTEN SCORES P2A	WRITTEN SCORES P2B	WRITTEN SCORES P3
4	1	4	79	18	32	29	39.0	5.0	4.2	10.1	9.7	10	10	9	78	24	15	5	34
4	1	7	73	15	36	22	45.0	10.7	8.3	15.6	9.3	10	9	9	96	26	18	12	40
4	1	10	54	14	28	12	51.0	11.9	12.0	24.4	8.0	9	9	6	45	14	5	0	26
4	1	16	52	16	24	12	39.0	10.7	6.2	3.6	7.0	9	6	6	75	26	5	10	34
4	1	22	51	8	22	21	41.0	2.8	1.5	1.9	5.7	5	6	6	68	22	9	2	35
4	1	25	71	18	24	29	25.0	3.0	1.1	3.6	8.3	10	6	9	88	24	16	12	36
4	1	28	34	9	18	7	45.0	15.8	4.5	13.0	8.0	9	6	9	46	17	1	4	24
4	1	31	38	8	30	0	42.0	12.1	4.8	0.1	7.7	9	8	6	78	23	18	3	34
4	2	43	85	19	30	36	43.0	7.1	10.1	13.4	9.3	10	9	9	86	30	14	5	37
4	2	46	29	11	18	0	45.0	9.7	9.3	0.8	6.0	7	7	4	21	11	0	0	10
4	2	49	51	18	19	14	43.0	9.1	7.6	10.0	9.3	10	9	9	67	22	17	3	25
4	2	52	22	3	12	7	41.0	9.9	2.9	6.6	6.3	5	7	7	28	14	5	2	7
4	3	58	76	18	40	18	39.0	6.5	3.3	8.8	9.3	9	10	9	84	24	18	5	37
4	3	61	76	15	40	21	48.0	10.6	10.9	8.1	9.0	10	10	7	60	24	16	12	8
4	3	70	21	8	13	0	43.0	15.3	18.2	2.0	9.0	9	9	9	42	16	8	12	6
4	4	73	78	9	40	29	36.0	8.1	3.3	7.0	9.3	9	10	9	74	20	2	12	40
4	4	76	68	12	40	16	40.0	2.8	3.0	1.6	7.0	7	10	4	34	12	0	0	22
4	1	13	47	10	26	11	22.0	9.4	5.8	7.4	7.0	7	7	7	42	10	7	3	22
4	2	34	45	12	26	7	23.0	9.6	4.1	10.1	7.0	7	7	7	68	23	13	12	20
4	2	55	63	12	23	28	41.0	9.4	10.3	20.8	7.0	7	7	7	43	24	17	2	0
4	1	19	68	18	36	14	37.0	11.0	5.1	21.2	9.0	9	9	9	96	30	18	12	36
4	2	37	68	10	36	22	45.0	14.7	12.2	18.0	9.0	9	9	9	69	21	4	9	35
4	2	40	68	18	36	14	41.0	19.8	14.0	7.8	9.0	9	9	9	68	26	4	4	34
4	3	64	56	14	24	18	51.0	18.5	7.4	25.3	9.0	9	9	9	55	26	0	0	29
4	3	67	32	9	16	7	43.0	11.6	10.9	20.9	9.0	9	9	9	18	11	5	0	2

TABLE J.2 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
4	1	9	13	9.9
	2	9	32	7.7
	3	9	22	10.1
	1	10	18	5.0
	2	10	32	4.2
	3	10	0	1.0
7	1	9	14	10.4
	2	9	36	8.3
	3	9	0	15.6
	1	10	15	10.7
10	1	9	14	11.9
	2	9	28	12.0
	3	9	0	0.5
	2	10	0	2.0
	3	6	12	24.4
16	1	9	16	10.7
	2	9	0	3.1
	2	6	24	6.2
	1	10	13	4.9
	3	9	0	1.0
	3	6	12	3.5
	3	6	12	3.6
	1	10	0	1.3
	3	6	6	1.5
	3	3	0	1.4
	3	1	4	0.5
	3	4	0	1.0
25	1	9	14	4.7
	2	9	8	2.2
	3	9	22	3.6
	3	10	7	2.5
	1	10	18	3.0
	2	6	24	1.1
	3	7	21	3.5
	3	9	7	3.4
	3	6	18	1.5

TABLE J.3: SEQUENCE OF PROBLEMS WORKED BY TAILORED EXAM SUBJECTS

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
22	1	9	4	4.1
	1	6	0	0.6
	1	3	0	1.0
	1	1	2	0.9
	2	9	0	1.7
	1	4	2	2.6
	2	6	0	0.9
	3	9	0	7.0
	2	3	9	1.7
	3	6	6	3.4
	1	2	4	0.7
	2	6	22	1.5
	3	3	12	0.5
	3	6	18	1.9
	1	5	8	2.8
	2	9	15	3.2
	3	9	0	3.1
	2	8	9	3.4
28	1	9	9	15.8
	2	9	11	11.8
	3	9	0	13.0
	2	6	18	4.5
31	1	9	8	12.1
	2	9	29	14.7
	3	9	0	4.5
	2	10	0	0.3
	1	8	0	0.3
	2	7	19	5.3
	3	6	0	0.1
	2	8	30	4.8
43	1	9	14	12.4
	2	9	30	10.1
	3	9	36	13.4
	1	10	19	7.1

TABLE J.3 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
46	1	7	11	9.7
	2	7	18	9.3
	3	7	0	14.9
	1	9	0	1.0
	2	8	0	0.1
	3	4	0	0.8
	1	6	6	8.7
49	1	9	14	13.0
	1	10	18	9.1
	1	10	15	3.9
	2	9	19	7.6
	3	9	0	10.0
52	1	7	3	8.6
	1	4	2	3.3
	1	2	3	4.0
	2	7	12	2.9
	1	5	3	9.9
	2	6	11	2.1
	3	7	7	6.6
	2	6	8	1.3
	3	4	0	2.5
58	1	9	10	6.5
	2	9	36	7.5
	3	9	14	8.8
	3	7	7	2.9
	3	4	11	1.7
	2	10	40	4.4
	2	10	40	3.3
	1	10	17	4.2
61	1	7	14	9.3
	2	7	28	7.8
	3	7	21	8.1
	2	10	40	10.9
	3	9	0	0.1
	2	10	0	0.2
	1	10	15	10.6

TABLE J.3 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
70	1	9	8	15.3
	2	9	13	18.2
	1	8	0	1.9
	2	7	0	0.1
	3	9	0	2.0
	1	5	6	5.8
73	1	9	9	8.1
	2	9	36	7.5
	3	9	22	7.0
	2	10	33	3.7
	3	10	0	0.2
	3	7	21	3.0
	1	9	0	0.6
	2	10	40	3.3
	3	9	0	0.2
	2	10	0	0.4
	3	6	0	0.1
	2	8	0	1.8
76	1	7	10	6.0
	2	7	26	4.0
	3	7	0	5.3
	1	9	7	4.8
	2	10	36	4.9
	3	4	16	1.6
	1	7	12	2.8
	2	10	40	3.0
	3	7	7	3.0
	2	10	24	3.1
	3	4	0	1.8
13	1	7	10	9.4
	2	7	26	5.8
	3	7	7	7.4
19	1	9	18	11.0
	2	9	36	5.1
	3	9	14	21.2
34	1	7	12	9.6
	2	7	26	4.1
	3	7	7	10.1

TABLE J.3 (continued)

Subject Number	Problem Number	Diff. Level	Score	Time (min.)
37	1	9	10	14.7
	2	9	36	12.2
	3	9	7	18.0
40	1	9	18	19.8
	2	9	36	14.0
	3	9	14	7.8
55	1	7	12	9.4
	2	7	23	10.3
	3	7	28	20.8
64	1	9	14	18.5
	2	9	24	7.4
	3	9	14	25.3
67	1	9	9	11.6
	2	9	16	10.9
	3	9	0	20.9

TABLE J.3: (continued)

APPENDIX K: TABLES USED IN THE GENERATOR SECTIONS OF PROBLEM GENERATOR/GRADERS

The tables used to guide the generation of problems in the READ with FORMAT pg/g, the One-dimensional Fortran Arrays pg/g, and the DO-loops Over An Expression pg/g are given below. The tables for the Fortran Expressions pg/g and their use are discussed in Section 3.1.3.

In the READ with FORMAT pg/g, an instructor can select the concepts he wants tested. Table K.1 shows which concepts may be tested for each level of difficulty. For a given difficulty level a concept is tested if there is an "X" for that concept under the difficulty level number and if that concept was selected by the instructor. Table K.2 lists the problem complexity factors and Table K.3 lists the item complexity factors for each level of difficulty.

In the One-dimensional Fortran Arrays pg/g, there is no explicit selection of concepts as in the previously described pg/g's. Additional concepts are tested as problem difficulty is increased. Table K.4 lists the complexity factors for this pg/g.

In the DO-loops Over An Expression pg/g, an instructor may choose the languages PL/1 or Fortran. Table K.5 lists the complexity factors for this pg/g.

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Concepts:										
I format[1]	X	X	X	X	X	X	X	X	X	X
X format		X	X	X	X	X	X	X	X	X
F format				X	X	X	X	X	X	X
E format						X	X	X	X	X
field count			X		X		X	X		X
group count									X	X

note 1: I format is included by default at all levels of difficulty.

TABLE K.1: CONCEPTS WHICH MAY BE TESTED IN A READ WITH FORMAT PROBLEM FOR EACH LEVEL OF DIFFICULTY

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Complexity Factors:										
number of variables:	2	2	2	3	3	4	4	5	5	6
number of characters in the variable names:	1	1	2	2	2	3	3	2-3	2-3	2-3
number of formats used										
I format:	2	2	2	2	1	2	1	1	1	2
X format:	-	2	2	3	3	3	3	4	4	4
F format:	-	-	-	1	2	1	2	2	2	2
E format:	-	-	-	-	-	1	2	2	2	2
format with which field count is used:	-	-	I	-	F	-	F	E	-	I
formats with which group count is used:	-	-	-	-	-	-	-	-	I,F X,X	F,E X,X
number of extra characters on input card:	0	0	0	1	1	1	2	2	2	2

TABLE K.2: PROBLEM COMPLEXITY FACTORS USED FOR EACH LEVEL OF DIFFICULTY IN THE READ WITH FORMAT PROBLEM GENERATOR/GRADER

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Complexity Factors:										
field width for I format:	1	1	1	2	2	2	3	3	4	4
count for X code:	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3
characters[1] on input card corresponding to X code:	b	dg	dg	dg	dg	dg	dg	dg	dg	dg
w in Fw.d:	-	-	-	3	4	4	5	6	7	8
d in Fw.d:	-	-	-	1-2	1-3	1-3	1-4	1-5	1-6	1-7
decimal point included on input card for F format:	-	-	-	yes	yes	yes	no	no	no	no
w in Ew.d:	-	-	-	-	-	6	7	8	9	9
d in Ew.d:	-	-	-	-	-	0-2	0-3	1-8	1-9	0-7
form of the exponent on input card for E format:	-	-	-	-	-	A[2]	A[2]	B[2]	B[2]	C[2]
decimal point included on input card for E format:	-	-	-	-	-	yes	no	yes	no	no

note 1: b=blank columns on input card; dg=a digit (0-9) used on input card.

note 2: exponent forms: A = "E+dg" or "E-dg"
B = no exponent included on input card
C = "+dg" or "-dg"
If the sign of the exponent is minus, then dg may range from 0 to 9-d; if the sign is plus, then dg may range from 0 to d+2.

TABLE K.3: ITEM COMPLEXITY FACTORS USED FOR EACH LEVEL OF DIFFICULTY IN THE READ WITH FORMAT PROBLEM GENERATOR/GRADER

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Complexity Factors:										
number of characters in array names:	1	1	1	1	1	1	1	1	2	2
number of arrays:	1	1	1	1	1	1	2	2	2	2
number of elements in each array:	3	3	3	4	4	4	4	5	5	5
IF-loop included in the program segment:	no	no	no	no	no	yes	yes	yes	yes	yes
means by which arrays are initialized[1]:	A	A	A	A	B	C	D	E	F	G
calculations performed[2]:	H	I	J	K	K	L	M	N	O	P

note 1: means by which arrays are initialized:
A = initial values displayed in the array.
B = values assigned in assignment statements.
C = values initialized in the type statement.
D = one array is initialized in the type statement, the other is initialized by assignment statements.
E = both arrays are initialized in the type statement.
F = both arrays are initialized in the type statement, one of the initializations uses replication factors.
G = both arrays are initialized in the type statement, both initializations use replication factors.

note 2: calculations performed in the program segment. "I" is the index variable used in the program segment.
H = assign one array element a new value; eg. A(2) = 7 .
I = perform a calculation on one element; eg. A(3) = A(3) * 3 .
J = perform a calculation on one element involving another element; eg. A(2) = 2 - A(1) .
K = two calculations of the style described in J.

TABLE K.4: COMPLEXITY FACTORS USED FOR EACH LEVEL OF DIFFICULTY IN THE ONE-DIMENSIONAL FORTRAN ARRAYS PROBLEM GENERATOR/GRADER

L = assign a new value to each element in the array;
eg. A(I) = 3 * I .
M = assign each element in one array a value calculated from one element of the other array;
eg. A(I) = B(4) + I .
N = assign each element in one array the value from an element in the other array; eg. A(I) = B(6-I) .
O = assign each element in one array a value calculated with the value from an element in the other array;
eg. A(I) = B(6-I) + 3 .
P = assign each element in one array a value calculated with the value from another element in that array and the value from an element in the other array;
eg. A(I) = B(I+1) - A(I-1) .

TABLE K.4 (continued)

Difficulty Level:	1	2	3	4	5	6	7	8	9	10
Complexity Factors:										
number of iterations:	2	3	3	4	4	4	4	5	5	5
IF and GOTO statements included in the program segment:	no	no	no	no	no	yes[1]	yes[1]	yes[2]	yes[2]	yes[2]
number of characters in the index variable name:	1[3]	1[3]	1[3]	1[3]	1[4]	1[4]	2[5]	2[5]	2[5]	2[5]
expression[6]:	A	A	A	B	B	B	B	C	C	C

note 1: The GOTO statement terminates the loop.

note 2: The GOTO statement causes one line of values not to be printed.

note 3: The one character used is "I".

note 4: The one character used is randomly selected from the letters "JKLMN".

note 5: The first character is a letter and the second character is a digit.

note 6: Expression used: below, "a" may range from 2 to 5, "b" may range from 3 to 9, "I" is the DO-loop index variable, and "T" is a temporary work variable.
A: T = a * I .
B: T = a * I + b , or T = a * I - b .
C: T = T + a * T + b , or T = T + a * T - b .

TABLE K.5: COMPLEXITY FACTORS USED FOR EACH LEVEL OF DIFFICULTY IN THE DO-LOOPS OVER AN EXPRESSION PROBLEM GENERATOR/GRADER

APPENDIX L: TYPICAL PROBLEMS PRODUCED BY THE GENERATIVE EXAM SYSTEM

This appendix contains examples of problems produced by each problem generator/grader used in the experiments conducted to evaluate the Generative Exam System.

Type in the value for each expression. Assume default declarations for the variables. Include a decimal point if and only if the value is real.

```
For  J=7  N=2  K=-7  calculate:
8 * J - N + K

For  L=18  M=2  J=9  calculate:
L / M / (8 - J)

For  R=-1.5  Z=8.  calculate:
IFIX(7. - R + Z)

For  J=35  M=-10  K=-2  calculate:
J / 9 + M - K

For  N=8  I=4  K=-3  calculate:
7. + N / I * K

For  N=-1  J=4  L=-10  calculate:
N * J + 7 - L
```

FIGURE L.1: TYPICAL PROBLEM ON FORTRAN EXPRESSIONS, LEVEL 5

```
Type in the value for each expression.  Assume default
     declarations for the variables.  Include a decimal
     point if and only if the value is real.

For  ME=1  L=-10  IY=2  calculate:
10 + ME * L / IY                                 >

For  IY=24  JU=8  ME=5  calculate:
IY / JU + 6. - ME

For  M=44  IY=9  L=-6  calculate:
M / IY * L + 1

For  M=0  IY=5  calculate:
IABS(M ** IY * 5)

For  JU=-6  calculate:
 - JU ** 2

For  M=5  L=-4  ME=-10  calculate:
M - ((L + ME) - 0)

For  IY=29  JU=6  L=5  calculate:
IY / JU / 1 ** L
```

FIGURE L.2: TYPICAL PROBLEM ON FORTRAN EXPRESSIONS, LEVEL 7

```
Type in the value for each expression.  Assume default
     declarations for the variables.  Include a decimal
     point if and only if the value is real.

For  CI=9.  VO=-4.  VEC=-10.  calculate:
CI*VO-VEC-10.-4.

For  CI=-10.  TUP=2.  VO=2.  calculate:
-CI**TUP-VO

For  SIB=4.  VEC=3.  VO=3.  calculate:
IFIX(SIB+4.**VEC*VO)

For  NI=18  SIB=9.  MAW=6  calculate:
NI/SIB+2.5+MAW-5.

For  SOB=-8.  VO=2.  VEC=2.  calculate:
SOB**1.**VO+VEC

For  VO=8.  SIB=4.  SOB=-2.  calculate:
VO/SIB/((6.-10.)-SOB)

For  LAC=8  NU=-1  IU=0  calculate:
79/LAC+NU-IU+10

For  NU=-3  NI=9  IU=9  calculate:
NU*NI+10*IU/4
```

FIGURE L.3:   TYPICAL PROBLEM ON FORTRAN EXPRESSIONS, LEVEL 9

Type in the value for each expression. Assume default declarations for the variables. Include a decimal point if and only if the value is real.

```
For  NOC=17  JET=3  LEB=1  LE=0  KIW=2  calculate:
NOC/JET*LEB**LE/KIW

For  SA=1.  DOF=5.  RUB=-3.  calculate:
ABS(7.+SA**DOF*RUB)

For  LE=0  JET=-5  LEB=-3  NOC=2  LIF=-1  calculate:
LE+(JET-(LEB+(NOC*LIF)))

For  SA=6.  PU=1.  FY=2.  DOF=-5.  calculate:
SA**PU**FY-DOF

For  SA=9.  RUB=8.  DOF=1.  PU=5.  FAR=6.  calculate:
SA+RUB**DOF*PU+FAR

For  FY=36.  DOF=9.  RUB=4.  PU=-6.  calculate:
FY/DOF/5/RUB+PU

For  DOF=-5.  RUB=0.  PU=2.  calculate:
-DOF**RUB*PU

For  NOC=-8  LEB=0  KIW=2  JET=-5  LE=3  calculate:
((NOC-LEB)*KIW)*(JET-LE)
```

FIGURE L.4: TYPICAL PROBLEM ON FORTRAN EXPRESSIONS, LEVEL 10

```
Type in what this Fortran segment prints.

Enter "end" when there is no more output to be printed.
Enter "del" to delete an answer.
```

<pre>   INTEGER K4,J
      DO 2Ø K4 = 1, 7, 2
      J = 4 * K4 - 9
      PRINT, K4, J
2Ø   CONTINUE</pre>
	OUTPUT: K4 J
	
	>

FIGURE L.5: TYPICAL PROBLEM ON DO-LOOPS, LEVEL 5

Type in what this Fortran segment prints.

Enter "end" when there is no more output to be printed.
Enter "del" to delete an answer.

```
      INTEGER P8,A8                       | OUTPUT:   P8              A8
      A8 = 0                              |_____________________________
      DO  20  P8 = 7, 29, 4               | >
      IF (P8 .EQ. 23)  GOTO 30            |
      A8 = 5 * P8 - 4                     |
      PRINT, P8, A8                       |
20    CONTINUE                            |
30    CONTINUE                            |
```

FIGURE L.6:   TYPICAL PROBLEM ON DO-LOOPS, LEVEL 7

Type in what this Fortran segment prints.

Enter "end" when there is no more output to be printed.
Enter "del" to delete an answer.

<pre>   INTEGER X2,E7,P4,U4,B7
      P4 = 5
      U4 = 27
      E7 = Ø
      B7 = 4
      DO 2Ø X2 = P4, U4, B7
      IF (X2 .EQ. 13) GOTO 2Ø
      E7 = E7 + 4 * X2 - 5
      PRINT, X2, E7
2Ø   CONTINUE
3Ø   CONTINUE</pre>
	OUTPUT: X2 E7
	
	>

FIGURE L.7: TYPICAL PROBLEM ON DO-LOOPS, LEVEL 9

Type in what this Fortran segment prints.

Enter "end" when there is no more output to be printed.
Enter "del" to delete an answer.

<pre>   INTEGER X6,G4,H8,A6,E6,T9
      E6 = 8
      H8 = 9
      T9 = 41
      A6 = 51
      G4 = Ø
      DO 2Ø X6 = H8, A6, E6
      IF (X6 .EQ. T9) GOTO 2Ø
      G4 = G4 + 4 * X6 - 5
      PRINT, X6, G4
2Ø   CONTINUE
3Ø   CONTINUE</pre>
	OUTPUT: X6 G4
	>

FIGURE L.8: TYPICAL PROBLEM ON DO-LOOPS, LEVEL 10

Show the values contained in array A after executing statement 1Ø and after executing statement 3Ø.

```
     INTEGER I,
    X  A(4)
     A(1)=-2
     A(2)=Ø
     A(3)=1
     A(4)=3
1Ø   CONTINUE
     A(3)=4*A(1)
     A(4)=2+A(2)
3Ø   CONTINUE
```

Values in array at statement 1Ø:

A(1)	A(2)	A(3)	A(4)

Values in array at statement 3Ø:

A(1)	A(2)	A(3)	A(4)

FIGURE L.9: TYPICAL PROBLEM ON ONE-DIMENSIONAL ARRAYS, LEVEL 5

Show the values contained in array B and array T after executing statement 10 and after executing statement 30.

```
      INTEGER I,
     X  B(4)/1,1,4,0/,
     X  T(4)/2,4,5,-3/
      T(1)=2
      T(2)=4
      T(3)=5
      T(4)=-3
10    CONTINUE
      I=1
20    CONTINUE
      B(I)=T(3)+I
      I=I+1
      IF (I.LE.4) GO TO 20
30    CONTINUE
```

Values in arrays at statement 10:

B(1)	B(2)	B(3)	B(4)

T(1)	T(2)	T(3)	T(4)

Values in arrays at statement 30:

B(1)	B(2)	B(3)	B(4)

T(1)	T(2)	T(3)	T(4)

FIGURE L.10: TYPICAL PROBLEM ON ONE-DIMENSIONAL ARRAYS, LEVEL 7

Show the values contained in array RE and array TO after executing statement 10 and after executing statement 30.

```
      INTEGER I,
     X  RE(5)/4*2,0/,
     X  TO(5)/3*3,2*0/
10    CONTINUE
      I=1
20    CONTINUE
      RE(I)=TO(I+1)-RE(6-I)
      I=I+1
      IF (I.LE.4) GO TO 20
30    CONTINUE
```

Values in arrays at statement 10:

RE(1)	RE(2)	RE(3)	RE(4)	RE(5)

TO(1)	TO(2)	TO(3)	TO(4)	TO(5)

Values in arrays at statement 30:

RE(1)	RE(2)	RE(3)	RE(4)	RE(5)

TO(1)	TO(2)	TO(3)	TO(4)	TO(5)

FIGURE L.11: TYPICAL PROBLEM ON ONE-DIMENSIONAL ARRAYS, LEVEL 10

Type in the value stored in each variable by the following program segment. Include a decimal point if and only if the value is real.

```
   READ10,KE,TA,FO
10 FORMAT(I2,3X,2X,1X,2F4.1)
```

Input Card

```
1 Column 10        20        30        40        50        60
↓        ↓         ↓         ↓         ↓         ↓         ↓
95       758.067.7
```

KE=

TA=

FO=

FIGURE L.12: TYPICAL PROBLEM ON FORTRAN READ WITH FORMAT, LEVEL 5

Type in the value stored in each variable by the following program segment. Include a decimal point if and only if the value is real.

```
   READ10,WEB,PIM,VAD,NOG
10 FORMAT(E7.0,3X,3X,2F5.3,1X,I3)
```

Input Card

```
1 Column 10        20        30        40        50        60
6233E-141878408442025810155 66
```

WEB=

PIM=

VAD=

NOG=

FIGURE L.13: TYPICAL PROBLEM ON FORTRAN READ WITH FORMAT, LEVEL 7

Type in the value stored in each variable by the following program segment. Include a decimal point if and only if the value is real.

```
      READ10,WOF,JED,RA,MAF,VOR
   10 FORMAT(1X,3X,2(E9.3,I4,3X,1X),E9.9)
```

Input Card

```
8439027270317696 924863580921929 1462780711789855
```

WOF=

JED=

RA=

MAF=

VOR=

FIGURE L.14: TYPICAL PROBLEM ON FORTRAN READ WITH FORMAT, LEVEL 9

VITA

Lawrence Robert Whitlock was born September 7, 1946 in St. Louis, Missouri. He graduated from Downers Grove Community High School, Downers Grove, Illinois, in 1964. While attending Miami University as an undergraduate he was elected to Phi Beta Kappa, Phi Eta Sigma, and Phi Mu Alpha. In 1968 he graduated magna cum laude and with General Honors with a Bachelor of Music degree having concentrated in organ performance. At the University of Illinois, he has received the Master of Science degree in Computer Science (1974) and served as a research assistant for four years in the Department of Computer Science.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-76-821	2.	3. Recipient's Accession No.
4. Title and Subtitle Interactive Test Construction and Administration in the Generative Exam System			5. Report Date August 30, 1976
			6.
7. Author(s) Lawrence Robert Whitlock			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. NSF EPP 7421590
12. Sponsoring Organization Name and Address National Science Foundation 1800 G Street, N.W. Washington, D.C. 20550			13. Type of Report & Period Covered Ph.D. Dissertation
			14.
15. Supplementary Notes			

16. Abstracts

This thesis describes the design, implementation, and evaluation of the Generative Exam System, a completely interactive system for the construction and administration of examinations. The heart of the system is a set of problem generator/grader modules which generate, administer, grade, and review examination problems with students.

A tailored style examination is introduced in which the difficulty levels of the problems are altered as the student works through the exam in an attempt to match the problem difficulty level to the student's level of knowledge. Experiments conducted to evaluate the Generative Exam System indicate that examinations administered by the system are as effective at evaluating students as written exams.

17. Key Words and Document Analysis. 17a. Descriptors

Generative Exam System

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release Unlimited	UNCLASSIFIED	164
	20. Security Class (This Page) UNCLASSIFIED	22. Price -----

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71